MFSR Chinook Spawn Timing Phenology

2025-04-23

Methods

In this paper, we 1)compare detailed salmon spawn timing data across four years in nine stream reaches within six major watersheds, all within a relatively intact river basin in central Idaho, and 2) examine how water temperature profiles, habitat features, and spawning escapements influence wild Chinook salmon phenotypic diversity, specifically, the timing of spawning.

Study area

(Text below verbatim from Isaak and Thurow 2006)

This study was conducted in the Middle Fork of the Salmon River (MFSR) in central Idaho (Fig. 1). The MFSR drains 7330 km2 of forested and steeply mountainous terrain in central Idaho that ranges in elevation from 1000 to 3150 m. Most of the area (>95%) is administered by the USDA Forest Service and was managed as a primitive area from 1930 to 1980 before receiving permanent protection as part of the Frank Church – River of No Return – Wilderness in 1980. As a result, road and trail densities are low and most areas exist in relatively pristine condition. Some areas continue to recover from the effects of grazing or mining, but cessation of many of these activities has occurred since wilderness designation and listing of Snake River salmon stocks under the Endangered Species Act. Natural disturbances from fires, hillslope movements, and floods persist, and these processes maintain a dynamic mosaic of landscape conditions.

More details. . .

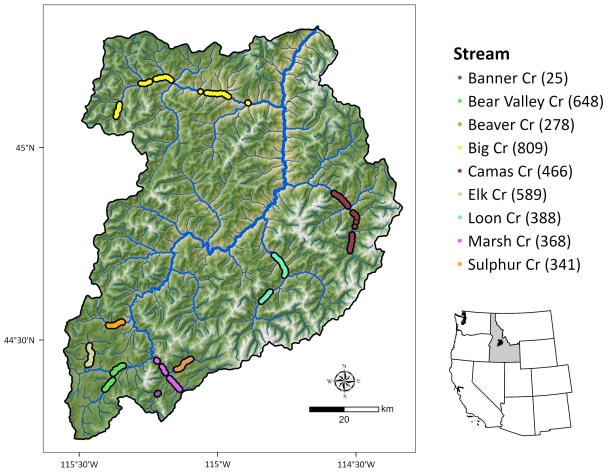


Figure 1: Map of the Middle Fork Salmon River (MFSR) study area showing redd locations used in the analysis (2002-2005) and stream reaches.

Spawn timing data

Spawn timing data for Chinook salmon were collected from 2001 to 2005 in the MFSR. We removed data from 2001, and data from Knapp Creek and Cape Horn Creek, as these sites were not consistently sampled.

We will use the `yday` column (day of year a redd is assumed to be complete) as our response variable (spawn timing). We visualize variation in day of year redds are complete using destiny plots:

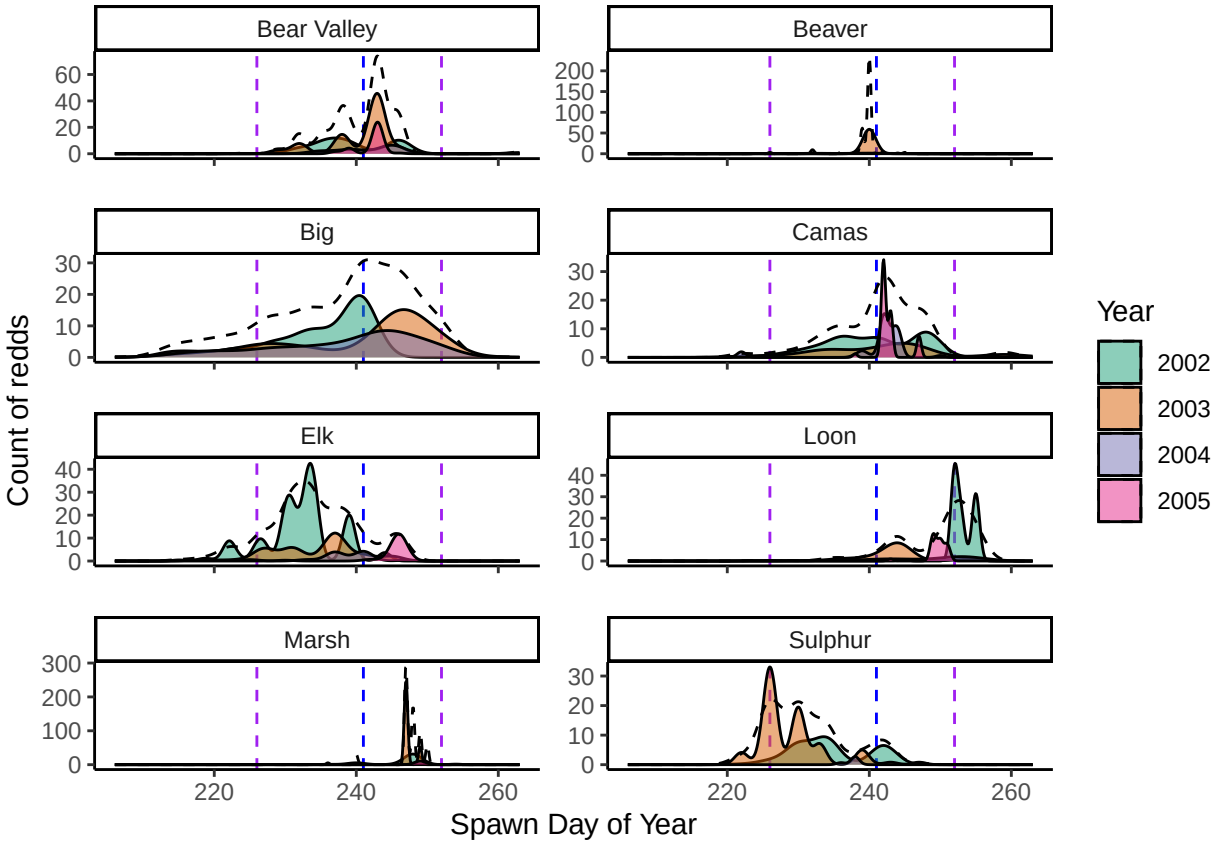


Figure 2: Temporal distribution of spawn timing for Chinook salmon in the MFSR. 2002-2005.

- colors = year
- x axis = day of year
- y-axis = count of unique redds
- dashed lines (black) = average spawning distribution by site across all years
- vertical dotted lines (purple) = 5th and 95th quantile for ALL MFSR redds across years
- vertical dotted line (blue): median (50th quantile) for ALL MFSR redds across years

Proportional cumulative redds

Next, for each year and stream system, we calculate the proportional cumulative number of redds. This is done by first calculating the cumulative number of redds for each year and stream, and then dividing that by the maximum cumulative number of redds for that year and stream.

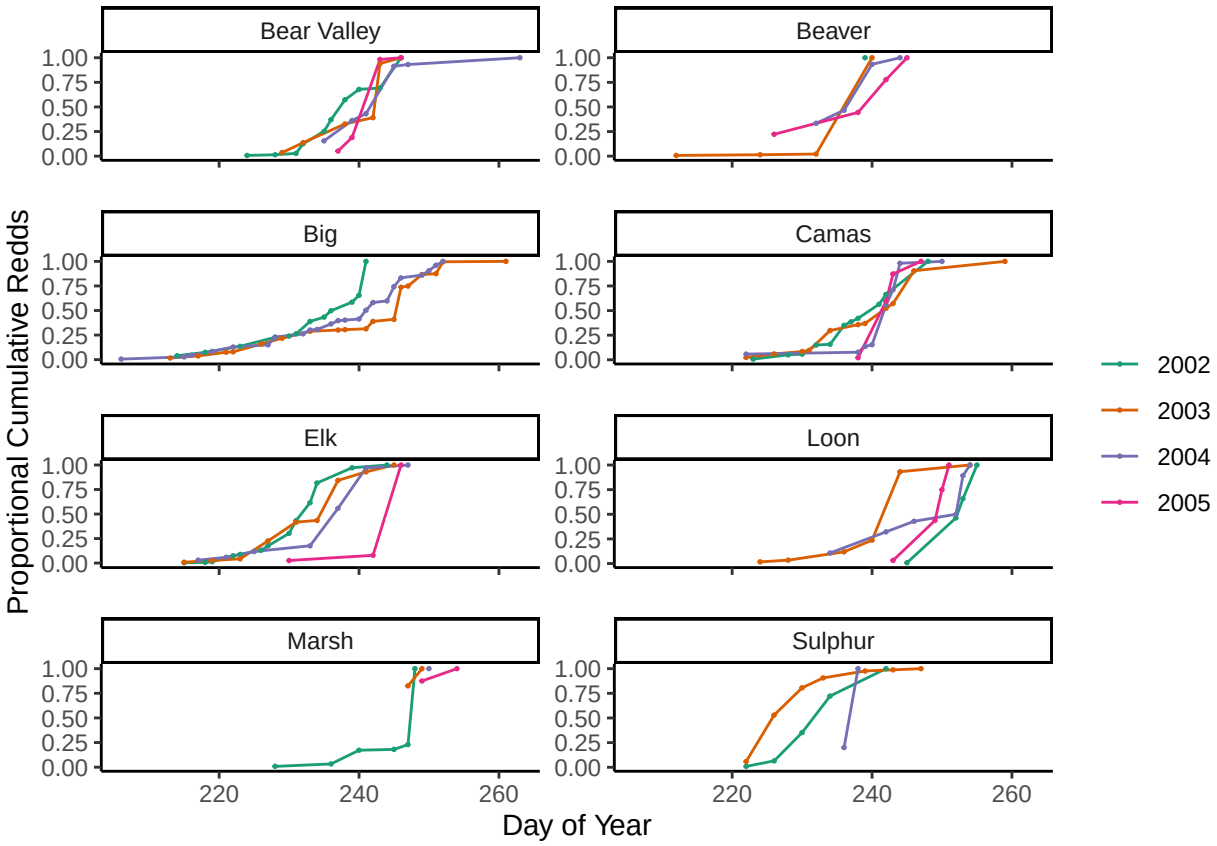


Figure 3: Proportional cumulative redds by stream.

- color = year
- x-axis = day of year
- y-axis = Proportional cumulative redds

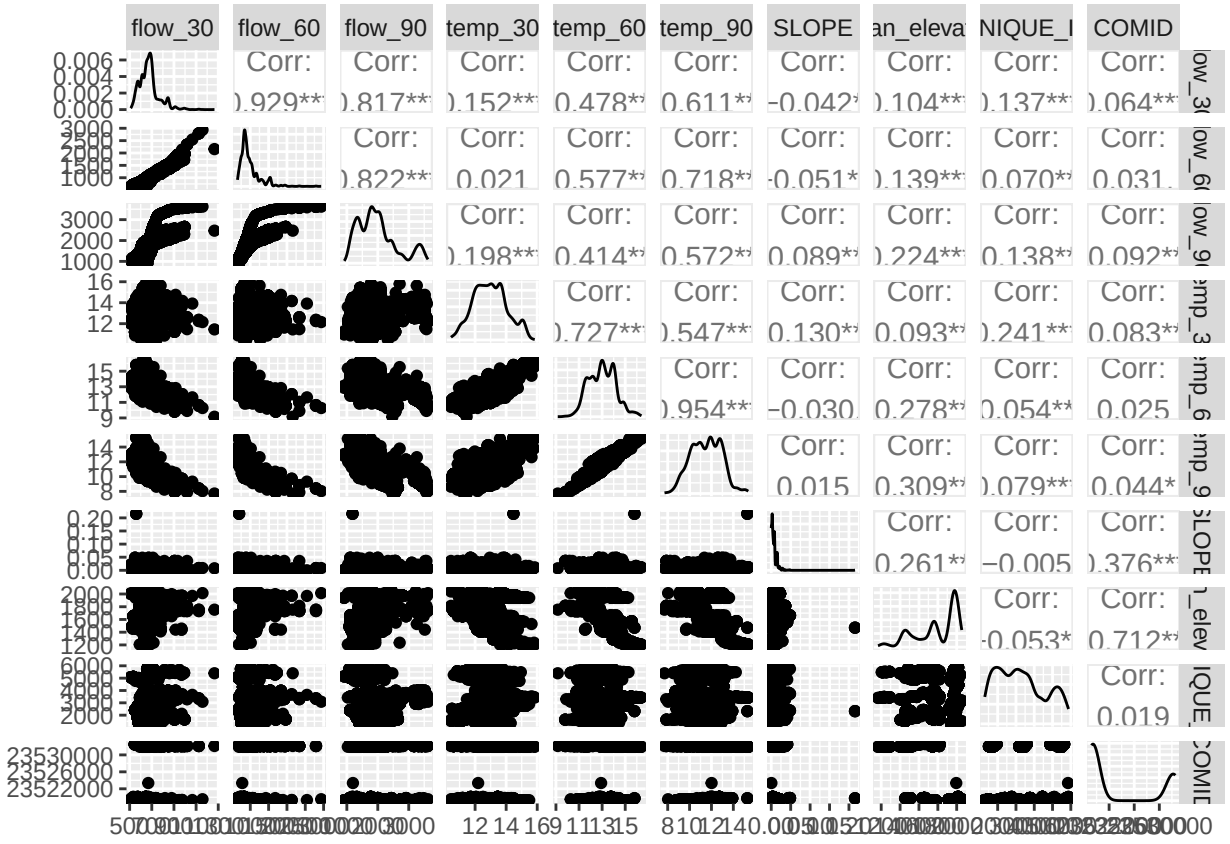
Models and model selection

Linear models

Next we load covariate data and combine with spawn timing data:

```
## # A tibble: 6 x 13
##    yday stream     year  flow_30 flow_60 flow_90 temp_30 temp_60 temp_90   SLOPE
##   <dbl> <chr>      <fct>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1   235 Bear Vall~ 2002     651.    997.   1993.    12.7    12.9    11.1 0.00299
## 2   235 Bear Vall~ 2002     651.    997.   1993.    12.7    12.9    11.1 0.00299
## 3   235 Bear Vall~ 2002     651.    997.   1993.    14.0    14.2    12.2 0.00172
## 4   235 Bear Vall~ 2002     651.    997.   1993.    14.0    14.2    12.2 0.00172
## 5   235 Bear Vall~ 2002     651.    997.   1993.    14.0    14.2    12.2 0.00172
## 6   235 Bear Vall~ 2002     651.    997.   1993.    13.8    14.0    11.9 0.00286
## # i 3 more variables: mean_elevation <dbl>, UNIQUE_ID <dbl>, COMID <dbl>
```

Now we check for colinearity between covariates (expect temp and flow to be bad). Remove covariates with corr >= 0.6.



Temperature and flow at the 30, 60, and 90 day intervals are colinear with one another within the larger covariate (temp or flow). So we need to choose our best predictive variable from that set (though note corr of temp_30 and temp_90 are <0.6, barely).

We also see high correlation between temp_90 and the different flow metrics. So we will look to see which temp metric to choose and make decisions about which flow metrics to include after.

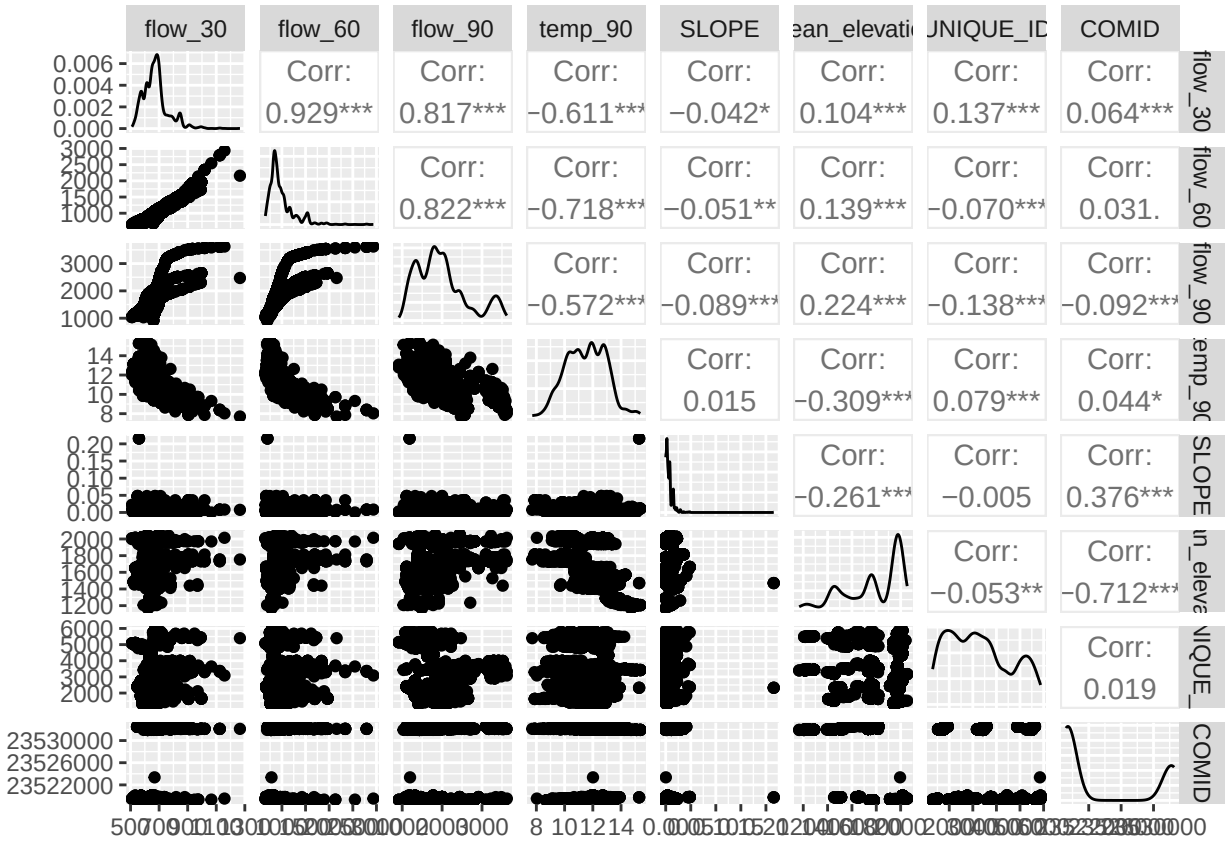
```
lm_30 <- lm(yday ~ temp_30, data = model_data)
lm_60 <- lm(yday ~ temp_60, data = model_data)
lm_90 <- lm(yday ~ temp_90, data = model_data)

AIC(lm_30, lm_60, lm_90)
```

```
##        df      AIC
## lm_30   3 21349.49
## lm_60   3 19738.48
## lm_90   3 18003.79
```

We can see the model with 90 days of temperature before spawning is much better than the other models (delta AIC = 1734.69)

Let's look at correlation again when removing temp_60 and temp_30.



So now we see that flow_30 and flow_60 are highly correlated with temp_90 so we need to also remove those.

Our final model covariates will be temp_90, slope, elevation, year, site, and abundance (once added). We next z-score the continuous covariates and fit three models:

1. Full model with all covariates
2. Model with interaction between stream and temp_90
3. Intercept only model

```
##                    df       AIC
```

```
## full_model      16 14264.37
## int_model       23 13662.93
## intercept_model  2 21367.91
```

From AIC selection we see that the interaction model preforms the best by far. We can look at the model output below.

```
##
## Call:
## lm(formula = yday ~ year + flow_90 + temp_90 * stream + SLOPE +
##     mean_elevation, data = model_data_final)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.9888  -0.9675   0.2593   1.4320  13.6831
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            234.32938    0.24895 941.271  < 2e-16 ***
## year2003                 5.38151    0.24004  22.419  < 2e-16 ***
## year2004                -0.62055    0.15453  -4.016 6.08e-05 ***
## year2005                -0.78716    0.20140  -3.908 9.49e-05 ***
## flow_90                 -6.51185    0.14034 -46.401  < 2e-16 ***
## temp_90                  2.17725    0.21667  10.049  < 2e-16 ***
## streamBeaver             4.40139    0.44221   9.953  < 2e-16 ***
## streamBig                6.56604    0.60065  10.931  < 2e-16 ***
## streamCamas              5.46363    0.42747  12.781  < 2e-16 ***
## streamElk                0.60421    0.20067   3.011  0.00263 **
## streamLoon               8.28262    0.51204  16.176  < 2e-16 ***
## streamMarsh              3.03676    0.27266  11.138  < 2e-16 ***
## streamSulphur            6.47217    0.63279  10.228  < 2e-16 ***
## SLOPE                   -0.02845    0.04980  -0.571  0.56780
## mean_elevation           3.78439    0.27815  13.605  < 2e-16 ***
## temp_90:streamBeaver     3.84665    0.53895   7.137 1.19e-12 ***
## temp_90:streamBig        3.40988    0.23547  14.481  < 2e-16 ***
## temp_90:streamCamas      1.12956    0.22748   4.965 7.24e-07 ***
## temp_90:streamElk        4.18607    0.29141  14.365  < 2e-16 ***
## temp_90:streamLoon       1.81809    0.37213   4.886 1.08e-06 ***
## temp_90:streamMarsh     -3.12560    0.35759  -8.741  < 2e-16 ***
## temp_90:streamSulphur    0.99663    0.41245   2.416  0.01574 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.321 on 2994 degrees of freedom
## Multiple R-squared:  0.9234, Adjusted R-squared:  0.9228
## F-statistic:  1718 on 21 and 2994 DF,  p-value: < 2.2e-16
```

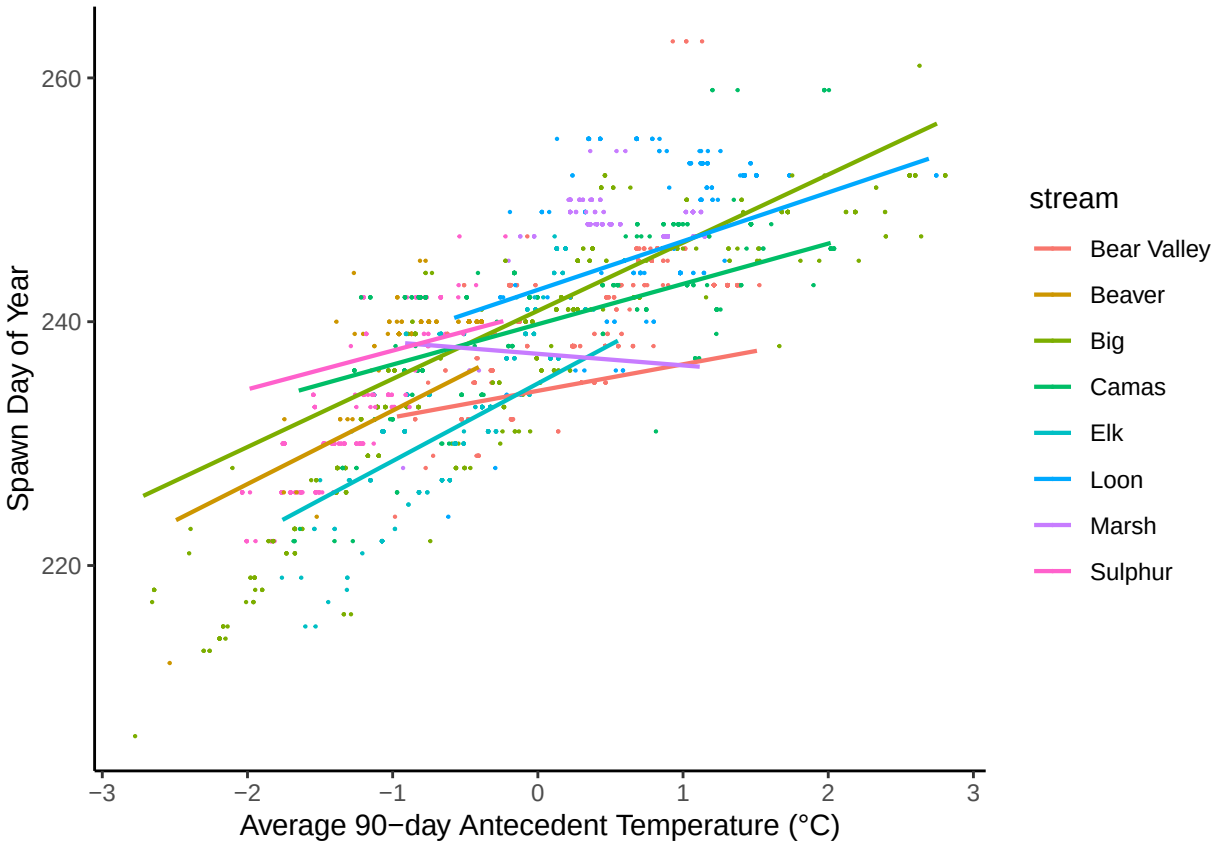
Marginal means Marginal means of stream:

```
## Estimated Marginal Means
##
## stream      |   Mean |   SE |           95% CI | t(2994)
## ------------------------------------------------------
```

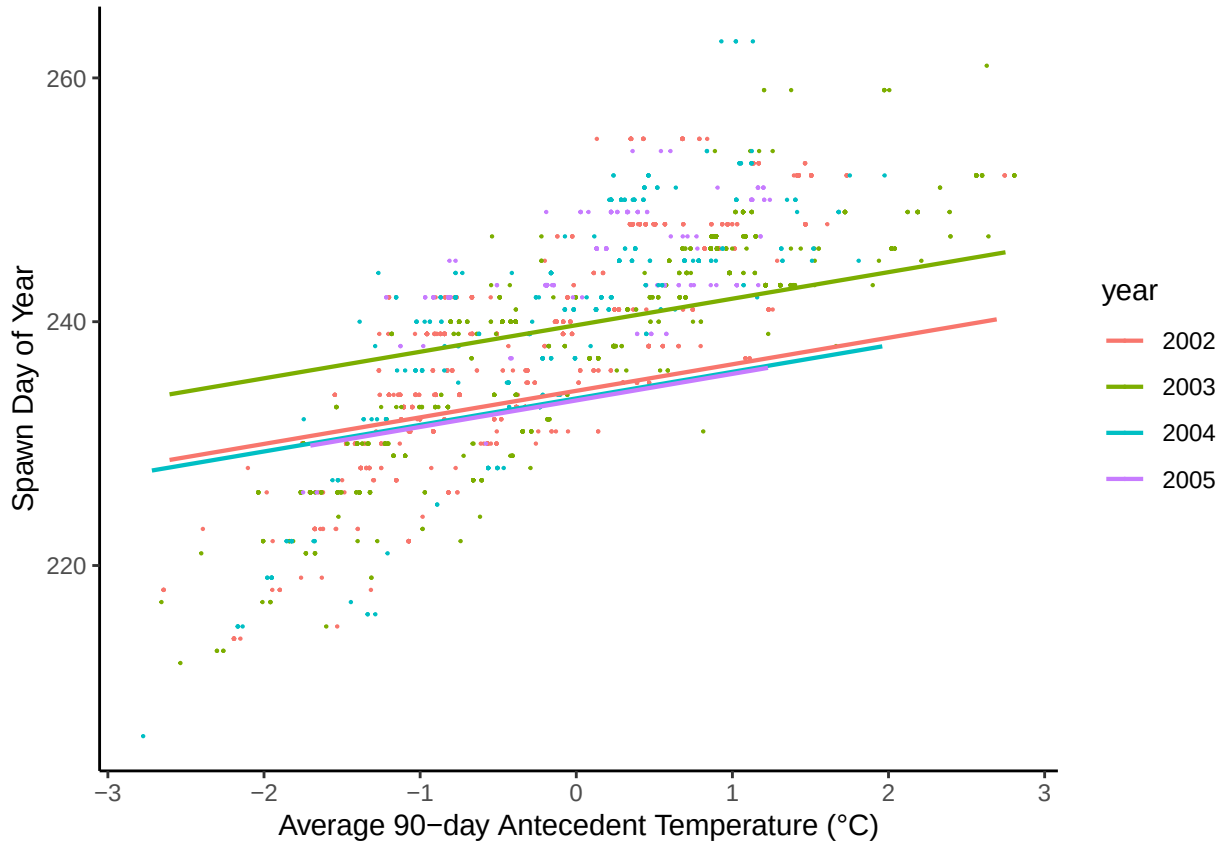
```
## Bear Valley | 235.32 | 0.25 | [234.84, 235.81] |  955.68
## Beaver      | 239.72 | 0.45 | [238.83, 240.62] |  526.98
## Big         | 241.89 | 0.39 | [241.13, 242.65] |  624.27
## Camas       | 240.79 | 0.23 | [240.34, 241.23] | 1061.03
## Elk         | 235.93 | 0.26 | [235.42, 236.44] |  910.45
## Loon        | 243.61 | 0.37 | [242.88, 244.33] |  658.52
## Marsh       | 238.36 | 0.32 | [237.72, 239.00] |  734.55
## Sulphur     | 241.80 | 0.55 | [240.72, 242.87] |  441.07
##
## Variable predicted: yday
## Predictors modulated: stream
## Predictors averaged: year, flow_90 (3.8e-16), temp_90 (-1.2e-15), SLOPE (-1.7e-16), mean_elevation (

## Estimated Marginal Means
##
## year |   Mean |   SE |          95% CI | t(2994)
## ------------------------------------------------
## 2002 | 238.68 | 0.14 | [238.41, 238.96] | 1689.37
## 2003 | 244.06 | 0.18 | [243.70, 244.42] | 1327.36
## 2004 | 238.06 | 0.20 | [237.66, 238.46] | 1161.32
## 2005 | 237.90 | 0.25 | [237.41, 238.38] |  958.16
##
## Variable predicted: yday
## Predictors modulated: year
## Predictors averaged: flow_90 (3.8e-16), temp_90 (-1.2e-15), stream, SLOPE (-1.7e-16), mean_elevation
```

Plot model fit of spawn date as a function of stream and temp_90 while holding other variables fixed:



Same but replace stream with year:



Mixed-effects model

Because the data are hierarchical, it makes sense to run the model in a mixed-model framework. It will essentially be the same model but with a random effect (slopes and intercepts) of Redd ID nested in COMID nested in stream nested in year.

We'll use a similar model selection process as before.

```
##                         df       AIC
## full_mod_mixed          17 13278.650
## int_mod_mixed           24 12573.751
## int_mod_mixed2          27  8032.027
## intercept_model_mixed    3 18629.305
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: yday ~ flow_90 * year + temp_90 * stream + SLOPE + mean_elevation +
##     (1 | COMID)
##    Data: model_data_final
##
##      AIC      BIC   logLik deviance df.resid
##   8032.0   8194.3  -3989.0   7978.0     2989
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -8.0971 -0.3838 -0.1077  0.3262  8.1897
```

```
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  COMID    (Intercept) 2.5688   1.6028
##  Residual             0.7212   0.8492
## Number of obs: 3016, groups:  COMID, 104
##
## Fixed effects:
##                          Estimate Std. Error t value
## (Intercept)           229.45703    0.57665 397.913
## flow_90                -9.78213    0.10214 -95.774
## year2003                4.38656    0.14092  31.128
## year2004               -5.52581    0.12016 -45.989
## year2005               -4.41371    0.30358 -14.539
## temp_90                 5.07231    0.18051  28.099
## streamBeaver            5.73495    0.82396   6.960
## streamBig              15.78216    1.14461  13.788
## streamCamas            11.57048    0.88345  13.097
## streamElk               3.94440    0.83220   4.740
## streamLoon             13.19760    0.94751  13.929
## streamMarsh             0.89254    0.68201   1.309
## streamSulphur          14.10488    0.83045  16.985
## SLOPE                  -0.21457    0.07230  -2.968
## mean_elevation          7.19557    0.39196  18.358
## flow_90:year2003        5.16290    0.06226  82.922
## flow_90:year2004       -4.68430    0.11758 -39.840
## flow_90:year2005       -1.73831    0.26694  -6.512
## temp_90:streamBeaver    1.01661    0.22823   4.454
## temp_90:streamBig       2.07835    0.14680  14.157
## temp_90:streamCamas     0.77884    0.18363   4.241
## temp_90:streamElk       1.45112    0.17094   8.489
## temp_90:streamLoon     -0.36264    0.25445  -1.425
## temp_90:streamMarsh     0.32502    0.20342   1.598
## temp_90:streamSulphur   1.14073    0.22179   5.143
```

Bayesian fit

Fit it bayesian to avoid all the singularity mumbo jumbo

```
## Running MCMC with 4 parallel chains...
##
## Chain 1 Iteration:    1 / 6000 [  0%]  (Warmup)
## Chain 2 Iteration:    1 / 6000 [  0%]  (Warmup)
## Chain 3 Iteration:    1 / 6000 [  0%]  (Warmup)
## Chain 4 Iteration:    1 / 6000 [  0%]  (Warmup)
## Chain 2 Iteration:  100 / 6000 [  1%]  (Warmup)
## Chain 1 Iteration:  100 / 6000 [  1%]  (Warmup)
## Chain 3 Iteration:  100 / 6000 [  1%]  (Warmup)
## Chain 4 Iteration:  100 / 6000 [  1%]  (Warmup)
## Chain 2 Iteration:  200 / 6000 [  3%]  (Warmup)
## Chain 3 Iteration:  200 / 6000 [  3%]  (Warmup)
## Chain 1 Iteration:  200 / 6000 [  3%]  (Warmup)
## Chain 4 Iteration:  200 / 6000 [  3%]  (Warmup)
```

```
## Chain 3 Iteration:  300 / 6000 [  5%]  (Warmup)
## Chain 3 Iteration:  400 / 6000 [  6%]  (Warmup)
## Chain 2 Iteration:  300 / 6000 [  5%]  (Warmup)
## Chain 1 Iteration:  300 / 6000 [  5%]  (Warmup)
## Chain 3 Iteration:  500 / 6000 [  8%]  (Warmup)
## Chain 4 Iteration:  300 / 6000 [  5%]  (Warmup)
## Chain 3 Iteration:  600 / 6000 [ 10%]  (Warmup)
## Chain 3 Iteration:  700 / 6000 [ 11%]  (Warmup)
## Chain 3 Iteration:  800 / 6000 [ 13%]  (Warmup)
## Chain 2 Iteration:  400 / 6000 [  6%]  (Warmup)
## Chain 3 Iteration:  900 / 6000 [ 15%]  (Warmup)
## Chain 1 Iteration:  400 / 6000 [  6%]  (Warmup)
## Chain 3 Iteration: 1000 / 6000 [ 16%]  (Warmup)
## Chain 3 Iteration: 1001 / 6000 [ 16%]  (Sampling)
## Chain 4 Iteration:  400 / 6000 [  6%]  (Warmup)
## Chain 3 Iteration: 1100 / 6000 [ 18%]  (Sampling)
## Chain 3 Iteration: 1200 / 6000 [ 20%]  (Sampling)
## Chain 2 Iteration:  500 / 6000 [  8%]  (Warmup)
## Chain 3 Iteration: 1300 / 6000 [ 21%]  (Sampling)
## Chain 1 Iteration:  500 / 6000 [  8%]  (Warmup)
## Chain 3 Iteration: 1400 / 6000 [ 23%]  (Sampling)
## Chain 4 Iteration:  500 / 6000 [  8%]  (Warmup)
## Chain 3 Iteration: 1500 / 6000 [ 25%]  (Sampling)
## Chain 3 Iteration: 1600 / 6000 [ 26%]  (Sampling)
## Chain 3 Iteration: 1700 / 6000 [ 28%]  (Sampling)
## Chain 2 Iteration:  600 / 6000 [ 10%]  (Warmup)
## Chain 1 Iteration:  600 / 6000 [ 10%]  (Warmup)
## Chain 3 Iteration: 1800 / 6000 [ 30%]  (Sampling)
## Chain 4 Iteration:  600 / 6000 [ 10%]  (Warmup)
## Chain 3 Iteration: 1900 / 6000 [ 31%]  (Sampling)
## Chain 3 Iteration: 2000 / 6000 [ 33%]  (Sampling)
## Chain 3 Iteration: 2100 / 6000 [ 35%]  (Sampling)
## Chain 2 Iteration:  700 / 6000 [ 11%]  (Warmup)
## Chain 1 Iteration:  700 / 6000 [ 11%]  (Warmup)
## Chain 3 Iteration: 2200 / 6000 [ 36%]  (Sampling)
## Chain 3 Iteration: 2300 / 6000 [ 38%]  (Sampling)
## Chain 4 Iteration:  700 / 6000 [ 11%]  (Warmup)
## Chain 3 Iteration: 2400 / 6000 [ 40%]  (Sampling)
## Chain 3 Iteration: 2500 / 6000 [ 41%]  (Sampling)
## Chain 2 Iteration:  800 / 6000 [ 13%]  (Warmup)
## Chain 3 Iteration: 2600 / 6000 [ 43%]  (Sampling)
## Chain 1 Iteration:  800 / 6000 [ 13%]  (Warmup)
## Chain 3 Iteration: 2700 / 6000 [ 45%]  (Sampling)
## Chain 4 Iteration:  800 / 6000 [ 13%]  (Warmup)
## Chain 3 Iteration: 2800 / 6000 [ 46%]  (Sampling)
## Chain 3 Iteration: 2900 / 6000 [ 48%]  (Sampling)
## Chain 3 Iteration: 3000 / 6000 [ 50%]  (Sampling)
## Chain 2 Iteration:  900 / 6000 [ 15%]  (Warmup)
## Chain 1 Iteration:  900 / 6000 [ 15%]  (Warmup)
## Chain 3 Iteration: 3100 / 6000 [ 51%]  (Sampling)
## Chain 3 Iteration: 3200 / 6000 [ 53%]  (Sampling)
## Chain 4 Iteration:  900 / 6000 [ 15%]  (Warmup)
## Chain 3 Iteration: 3300 / 6000 [ 55%]  (Sampling)
## Chain 2 Iteration: 1000 / 6000 [ 16%]  (Warmup)
```

```
## Chain 2 Iteration: 1001 / 6000 [ 16%]  (Sampling)
## Chain 3 Iteration: 3400 / 6000 [ 56%]  (Sampling)
## Chain 2 Iteration: 1100 / 6000 [ 18%]  (Sampling)
## Chain 2 Iteration: 1200 / 6000 [ 20%]  (Sampling)
## Chain 3 Iteration: 3500 / 6000 [ 58%]  (Sampling)
## Chain 4 Iteration: 1000 / 6000 [ 16%]  (Warmup)
## Chain 4 Iteration: 1001 / 6000 [ 16%]  (Sampling)
## Chain 1 Iteration: 1000 / 6000 [ 16%]  (Warmup)
## Chain 1 Iteration: 1001 / 6000 [ 16%]  (Sampling)
## Chain 2 Iteration: 1300 / 6000 [ 21%]  (Sampling)
## Chain 3 Iteration: 3600 / 6000 [ 60%]  (Sampling)
## Chain 4 Iteration: 1100 / 6000 [ 18%]  (Sampling)
## Chain 2 Iteration: 1400 / 6000 [ 23%]  (Sampling)
## Chain 3 Iteration: 3700 / 6000 [ 61%]  (Sampling)
## Chain 4 Iteration: 1200 / 6000 [ 20%]  (Sampling)
## Chain 2 Iteration: 1500 / 6000 [ 25%]  (Sampling)
## Chain 3 Iteration: 3800 / 6000 [ 63%]  (Sampling)
## Chain 2 Iteration: 1600 / 6000 [ 26%]  (Sampling)
## Chain 4 Iteration: 1300 / 6000 [ 21%]  (Sampling)
## Chain 3 Iteration: 3900 / 6000 [ 65%]  (Sampling)
## Chain 2 Iteration: 1700 / 6000 [ 28%]  (Sampling)
## Chain 4 Iteration: 1400 / 6000 [ 23%]  (Sampling)
## Chain 1 Iteration: 1100 / 6000 [ 18%]  (Sampling)
## Chain 3 Iteration: 4000 / 6000 [ 66%]  (Sampling)
## Chain 2 Iteration: 1800 / 6000 [ 30%]  (Sampling)
## Chain 4 Iteration: 1500 / 6000 [ 25%]  (Sampling)
## Chain 3 Iteration: 4100 / 6000 [ 68%]  (Sampling)
## Chain 2 Iteration: 1900 / 6000 [ 31%]  (Sampling)
## Chain 3 Iteration: 4200 / 6000 [ 70%]  (Sampling)
## Chain 4 Iteration: 1600 / 6000 [ 26%]  (Sampling)
## Chain 2 Iteration: 2000 / 6000 [ 33%]  (Sampling)
## Chain 3 Iteration: 4300 / 6000 [ 71%]  (Sampling)
## Chain 2 Iteration: 2100 / 6000 [ 35%]  (Sampling)
## Chain 4 Iteration: 1700 / 6000 [ 28%]  (Sampling)
## Chain 2 Iteration: 2200 / 6000 [ 36%]  (Sampling)
## Chain 3 Iteration: 4400 / 6000 [ 73%]  (Sampling)
## Chain 1 Iteration: 1200 / 6000 [ 20%]  (Sampling)
## Chain 4 Iteration: 1800 / 6000 [ 30%]  (Sampling)
## Chain 2 Iteration: 2300 / 6000 [ 38%]  (Sampling)
## Chain 3 Iteration: 4500 / 6000 [ 75%]  (Sampling)
## Chain 2 Iteration: 2400 / 6000 [ 40%]  (Sampling)
## Chain 4 Iteration: 1900 / 6000 [ 31%]  (Sampling)
## Chain 3 Iteration: 4600 / 6000 [ 76%]  (Sampling)
## Chain 2 Iteration: 2500 / 6000 [ 41%]  (Sampling)
## Chain 4 Iteration: 2000 / 6000 [ 33%]  (Sampling)
## Chain 3 Iteration: 4700 / 6000 [ 78%]  (Sampling)
## Chain 2 Iteration: 2600 / 6000 [ 43%]  (Sampling)
## Chain 4 Iteration: 2100 / 6000 [ 35%]  (Sampling)
## Chain 3 Iteration: 4800 / 6000 [ 80%]  (Sampling)
## Chain 1 Iteration: 1300 / 6000 [ 21%]  (Sampling)
## Chain 2 Iteration: 2700 / 6000 [ 45%]  (Sampling)
## Chain 3 Iteration: 4900 / 6000 [ 81%]  (Sampling)
## Chain 4 Iteration: 2200 / 6000 [ 36%]  (Sampling)
## Chain 2 Iteration: 2800 / 6000 [ 46%]  (Sampling)
```

```
## Chain 3 Iteration: 5000 / 6000 [ 83%]  (Sampling)
## Chain 4 Iteration: 2300 / 6000 [ 38%]  (Sampling)
## Chain 2 Iteration: 2900 / 6000 [ 48%]  (Sampling)
## Chain 3 Iteration: 5100 / 6000 [ 85%]  (Sampling)
## Chain 2 Iteration: 3000 / 6000 [ 50%]  (Sampling)
## Chain 4 Iteration: 2400 / 6000 [ 40%]  (Sampling)
## Chain 3 Iteration: 5200 / 6000 [ 86%]  (Sampling)
## Chain 2 Iteration: 3100 / 6000 [ 51%]  (Sampling)
## Chain 4 Iteration: 2500 / 6000 [ 41%]  (Sampling)
## Chain 3 Iteration: 5300 / 6000 [ 88%]  (Sampling)
## Chain 1 Iteration: 1400 / 6000 [ 23%]  (Sampling)
## Chain 2 Iteration: 3200 / 6000 [ 53%]  (Sampling)
## Chain 3 Iteration: 5400 / 6000 [ 90%]  (Sampling)
## Chain 2 Iteration: 3300 / 6000 [ 55%]  (Sampling)
## Chain 4 Iteration: 2600 / 6000 [ 43%]  (Sampling)
## Chain 2 Iteration: 3400 / 6000 [ 56%]  (Sampling)
## Chain 3 Iteration: 5500 / 6000 [ 91%]  (Sampling)
## Chain 4 Iteration: 2700 / 6000 [ 45%]  (Sampling)
## Chain 2 Iteration: 3500 / 6000 [ 58%]  (Sampling)
## Chain 3 Iteration: 5600 / 6000 [ 93%]  (Sampling)
## Chain 4 Iteration: 2800 / 6000 [ 46%]  (Sampling)
## Chain 2 Iteration: 3600 / 6000 [ 60%]  (Sampling)
## Chain 3 Iteration: 5700 / 6000 [ 95%]  (Sampling)
## Chain 4 Iteration: 2900 / 6000 [ 48%]  (Sampling)
## Chain 1 Iteration: 1500 / 6000 [ 25%]  (Sampling)
## Chain 2 Iteration: 3700 / 6000 [ 61%]  (Sampling)
## Chain 3 Iteration: 5800 / 6000 [ 96%]  (Sampling)
## Chain 2 Iteration: 3800 / 6000 [ 63%]  (Sampling)
## Chain 4 Iteration: 3000 / 6000 [ 50%]  (Sampling)
## Chain 3 Iteration: 5900 / 6000 [ 98%]  (Sampling)
## Chain 2 Iteration: 3900 / 6000 [ 65%]  (Sampling)
## Chain 4 Iteration: 3100 / 6000 [ 51%]  (Sampling)
## Chain 3 Iteration: 6000 / 6000 [100%]  (Sampling)
## Chain 3 finished in 95.0 seconds.
## Chain 2 Iteration: 4000 / 6000 [ 66%]  (Sampling)
## Chain 4 Iteration: 3200 / 6000 [ 53%]  (Sampling)
## Chain 2 Iteration: 4100 / 6000 [ 68%]  (Sampling)
## Chain 1 Iteration: 1600 / 6000 [ 26%]  (Sampling)
## Chain 2 Iteration: 4200 / 6000 [ 70%]  (Sampling)
## Chain 4 Iteration: 3300 / 6000 [ 55%]  (Sampling)
## Chain 2 Iteration: 4300 / 6000 [ 71%]  (Sampling)
## Chain 4 Iteration: 3400 / 6000 [ 56%]  (Sampling)
## Chain 2 Iteration: 4400 / 6000 [ 73%]  (Sampling)
## Chain 4 Iteration: 3500 / 6000 [ 58%]  (Sampling)
## Chain 2 Iteration: 4500 / 6000 [ 75%]  (Sampling)
## Chain 4 Iteration: 3600 / 6000 [ 60%]  (Sampling)
## Chain 2 Iteration: 4600 / 6000 [ 76%]  (Sampling)
## Chain 1 Iteration: 1700 / 6000 [ 28%]  (Sampling)
## Chain 2 Iteration: 4700 / 6000 [ 78%]  (Sampling)
## Chain 4 Iteration: 3700 / 6000 [ 61%]  (Sampling)
## Chain 2 Iteration: 4800 / 6000 [ 80%]  (Sampling)
## Chain 4 Iteration: 3800 / 6000 [ 63%]  (Sampling)
## Chain 2 Iteration: 4900 / 6000 [ 81%]  (Sampling)
## Chain 4 Iteration: 3900 / 6000 [ 65%]  (Sampling)
```

```
## Chain 2 Iteration: 5000 / 6000 [ 83%]  (Sampling)
## Chain 2 Iteration: 5100 / 6000 [ 85%]  (Sampling)
## Chain 4 Iteration: 4000 / 6000 [ 66%]  (Sampling)
## Chain 1 Iteration: 1800 / 6000 [ 30%]  (Sampling)
## Chain 2 Iteration: 5200 / 6000 [ 86%]  (Sampling)
## Chain 4 Iteration: 4100 / 6000 [ 68%]  (Sampling)
## Chain 2 Iteration: 5300 / 6000 [ 88%]  (Sampling)
## Chain 4 Iteration: 4200 / 6000 [ 70%]  (Sampling)
## Chain 2 Iteration: 5400 / 6000 [ 90%]  (Sampling)
## Chain 2 Iteration: 5500 / 6000 [ 91%]  (Sampling)
## Chain 4 Iteration: 4300 / 6000 [ 71%]  (Sampling)
## Chain 2 Iteration: 5600 / 6000 [ 93%]  (Sampling)
## Chain 1 Iteration: 1900 / 6000 [ 31%]  (Sampling)
## Chain 4 Iteration: 4400 / 6000 [ 73%]  (Sampling)
## Chain 2 Iteration: 5700 / 6000 [ 95%]  (Sampling)
## Chain 4 Iteration: 4500 / 6000 [ 75%]  (Sampling)
## Chain 2 Iteration: 5800 / 6000 [ 96%]  (Sampling)
## Chain 2 Iteration: 5900 / 6000 [ 98%]  (Sampling)
## Chain 4 Iteration: 4600 / 6000 [ 76%]  (Sampling)
## Chain 2 Iteration: 6000 / 6000 [100%]  (Sampling)
## Chain 2 finished in 121.6 seconds.
## Chain 4 Iteration: 4700 / 6000 [ 78%]  (Sampling)
## Chain 1 Iteration: 2000 / 6000 [ 33%]  (Sampling)
## Chain 4 Iteration: 4800 / 6000 [ 80%]  (Sampling)
## Chain 4 Iteration: 4900 / 6000 [ 81%]  (Sampling)
## Chain 4 Iteration: 5000 / 6000 [ 83%]  (Sampling)
## Chain 1 Iteration: 2100 / 6000 [ 35%]  (Sampling)
## Chain 4 Iteration: 5100 / 6000 [ 85%]  (Sampling)
## Chain 4 Iteration: 5200 / 6000 [ 86%]  (Sampling)
## Chain 4 Iteration: 5300 / 6000 [ 88%]  (Sampling)
## Chain 4 Iteration: 5400 / 6000 [ 90%]  (Sampling)
## Chain 1 Iteration: 2200 / 6000 [ 36%]  (Sampling)
## Chain 4 Iteration: 5500 / 6000 [ 91%]  (Sampling)
## Chain 4 Iteration: 5600 / 6000 [ 93%]  (Sampling)
## Chain 4 Iteration: 5700 / 6000 [ 95%]  (Sampling)
## Chain 1 Iteration: 2300 / 6000 [ 38%]  (Sampling)
## Chain 4 Iteration: 5800 / 6000 [ 96%]  (Sampling)
## Chain 4 Iteration: 5900 / 6000 [ 98%]  (Sampling)
## Chain 4 Iteration: 6000 / 6000 [100%]  (Sampling)
## Chain 4 finished in 145.7 seconds.
## Chain 1 Iteration: 2400 / 6000 [ 40%]  (Sampling)
## Chain 1 Iteration: 2500 / 6000 [ 41%]  (Sampling)
## Chain 1 Iteration: 2600 / 6000 [ 43%]  (Sampling)
## Chain 1 Iteration: 2700 / 6000 [ 45%]  (Sampling)
## Chain 1 Iteration: 2800 / 6000 [ 46%]  (Sampling)
## Chain 1 Iteration: 2900 / 6000 [ 48%]  (Sampling)
## Chain 1 Iteration: 3000 / 6000 [ 50%]  (Sampling)
## Chain 1 Iteration: 3100 / 6000 [ 51%]  (Sampling)
## Chain 1 Iteration: 3200 / 6000 [ 53%]  (Sampling)
## Chain 1 Iteration: 3300 / 6000 [ 55%]  (Sampling)
## Chain 1 Iteration: 3400 / 6000 [ 56%]  (Sampling)
## Chain 1 Iteration: 3500 / 6000 [ 58%]  (Sampling)
## Chain 1 Iteration: 3600 / 6000 [ 60%]  (Sampling)
## Chain 1 Iteration: 3700 / 6000 [ 61%]  (Sampling)
```

```
## Chain 1 Iteration: 3800 / 6000 [ 63%]  (Sampling)
## Chain 1 Iteration: 3900 / 6000 [ 65%]  (Sampling)
## Chain 1 Iteration: 4000 / 6000 [ 66%]  (Sampling)
## Chain 1 Iteration: 4100 / 6000 [ 68%]  (Sampling)
## Chain 1 Iteration: 4200 / 6000 [ 70%]  (Sampling)
## Chain 1 Iteration: 4300 / 6000 [ 71%]  (Sampling)
## Chain 1 Iteration: 4400 / 6000 [ 73%]  (Sampling)
## Chain 1 Iteration: 4500 / 6000 [ 75%]  (Sampling)
## Chain 1 Iteration: 4600 / 6000 [ 76%]  (Sampling)
## Chain 1 Iteration: 4700 / 6000 [ 78%]  (Sampling)
## Chain 1 Iteration: 4800 / 6000 [ 80%]  (Sampling)
## Chain 1 Iteration: 4900 / 6000 [ 81%]  (Sampling)
## Chain 1 Iteration: 5000 / 6000 [ 83%]  (Sampling)
## Chain 1 Iteration: 5100 / 6000 [ 85%]  (Sampling)
## Chain 1 Iteration: 5200 / 6000 [ 86%]  (Sampling)
## Chain 1 Iteration: 5300 / 6000 [ 88%]  (Sampling)
## Chain 1 Iteration: 5400 / 6000 [ 90%]  (Sampling)
## Chain 1 Iteration: 5500 / 6000 [ 91%]  (Sampling)
## Chain 1 Iteration: 5600 / 6000 [ 93%]  (Sampling)
## Chain 1 Iteration: 5700 / 6000 [ 95%]  (Sampling)
## Chain 1 Iteration: 5800 / 6000 [ 96%]  (Sampling)
## Chain 1 Iteration: 5900 / 6000 [ 98%]  (Sampling)
## Chain 1 Iteration: 6000 / 6000 [100%]  (Sampling)
## Chain 1 finished in 368.9 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 182.8 seconds.
## Total execution time: 368.9 seconds.

## Running MCMC with 4 parallel chains...
##
## Chain 1 Iteration:    1 / 6000 [  0%]  (Warmup)
## Chain 2 Iteration:    1 / 6000 [  0%]  (Warmup)
## Chain 3 Iteration:    1 / 6000 [  0%]  (Warmup)
## Chain 4 Iteration:    1 / 6000 [  0%]  (Warmup)
## Chain 3 Iteration:  100 / 6000 [  1%]  (Warmup)
## Chain 4 Iteration:  100 / 6000 [  1%]  (Warmup)
## Chain 1 Iteration:  100 / 6000 [  1%]  (Warmup)
## Chain 2 Iteration:  100 / 6000 [  1%]  (Warmup)
## Chain 3 Iteration:  200 / 6000 [  3%]  (Warmup)
## Chain 4 Iteration:  200 / 6000 [  3%]  (Warmup)
## Chain 1 Iteration:  200 / 6000 [  3%]  (Warmup)
## Chain 2 Iteration:  200 / 6000 [  3%]  (Warmup)
## Chain 3 Iteration:  300 / 6000 [  5%]  (Warmup)
## Chain 4 Iteration:  300 / 6000 [  5%]  (Warmup)
## Chain 1 Iteration:  300 / 6000 [  5%]  (Warmup)
## Chain 2 Iteration:  300 / 6000 [  5%]  (Warmup)
## Chain 3 Iteration:  400 / 6000 [  6%]  (Warmup)
## Chain 4 Iteration:  400 / 6000 [  6%]  (Warmup)
## Chain 1 Iteration:  400 / 6000 [  6%]  (Warmup)
## Chain 2 Iteration:  400 / 6000 [  6%]  (Warmup)
## Chain 4 Iteration:  500 / 6000 [  8%]  (Warmup)
## Chain 4 Iteration:  600 / 6000 [ 10%]  (Warmup)
## Chain 3 Iteration:  500 / 6000 [  8%]  (Warmup)
```

```
## Chain 1 Iteration:  500 / 6000 [  8%]  (Warmup)
## Chain 4 Iteration:  700 / 6000 [ 11%]  (Warmup)
## Chain 2 Iteration:  500 / 6000 [  8%]  (Warmup)
## Chain 4 Iteration:  800 / 6000 [ 13%]  (Warmup)
## Chain 4 Iteration:  900 / 6000 [ 15%]  (Warmup)
## Chain 3 Iteration:  600 / 6000 [ 10%]  (Warmup)
## Chain 4 Iteration: 1000 / 6000 [ 16%]  (Warmup)
## Chain 4 Iteration: 1001 / 6000 [ 16%]  (Sampling)
## Chain 1 Iteration:  600 / 6000 [ 10%]  (Warmup)
## Chain 2 Iteration:  600 / 6000 [ 10%]  (Warmup)
## Chain 4 Iteration: 1100 / 6000 [ 18%]  (Sampling)
## Chain 4 Iteration: 1200 / 6000 [ 20%]  (Sampling)
## Chain 4 Iteration: 1300 / 6000 [ 21%]  (Sampling)
## Chain 4 Iteration: 1400 / 6000 [ 23%]  (Sampling)
## Chain 3 Iteration:  700 / 6000 [ 11%]  (Warmup)
## Chain 1 Iteration:  700 / 6000 [ 11%]  (Warmup)
## Chain 4 Iteration: 1500 / 6000 [ 25%]  (Sampling)
## Chain 2 Iteration:  700 / 6000 [ 11%]  (Warmup)
## Chain 4 Iteration: 1600 / 6000 [ 26%]  (Sampling)
## Chain 4 Iteration: 1700 / 6000 [ 28%]  (Sampling)
## Chain 4 Iteration: 1800 / 6000 [ 30%]  (Sampling)
## Chain 3 Iteration:  800 / 6000 [ 13%]  (Warmup)
## Chain 4 Iteration: 1900 / 6000 [ 31%]  (Sampling)
## Chain 1 Iteration:  800 / 6000 [ 13%]  (Warmup)
## Chain 2 Iteration:  800 / 6000 [ 13%]  (Warmup)
## Chain 4 Iteration: 2000 / 6000 [ 33%]  (Sampling)
## Chain 4 Iteration: 2100 / 6000 [ 35%]  (Sampling)
## Chain 4 Iteration: 2200 / 6000 [ 36%]  (Sampling)
## Chain 3 Iteration:  900 / 6000 [ 15%]  (Warmup)
## Chain 4 Iteration: 2300 / 6000 [ 38%]  (Sampling)
## Chain 1 Iteration:  900 / 6000 [ 15%]  (Warmup)
## Chain 2 Iteration:  900 / 6000 [ 15%]  (Warmup)
## Chain 4 Iteration: 2400 / 6000 [ 40%]  (Sampling)
## Chain 4 Iteration: 2500 / 6000 [ 41%]  (Sampling)
## Chain 4 Iteration: 2600 / 6000 [ 43%]  (Sampling)
## Chain 3 Iteration: 1000 / 6000 [ 16%]  (Warmup)
## Chain 3 Iteration: 1001 / 6000 [ 16%]  (Sampling)
## Chain 4 Iteration: 2700 / 6000 [ 45%]  (Sampling)
## Chain 1 Iteration: 1000 / 6000 [ 16%]  (Warmup)
## Chain 1 Iteration: 1001 / 6000 [ 16%]  (Sampling)
## Chain 4 Iteration: 2800 / 6000 [ 46%]  (Sampling)
## Chain 2 Iteration: 1000 / 6000 [ 16%]  (Warmup)
## Chain 2 Iteration: 1001 / 6000 [ 16%]  (Sampling)
## Chain 4 Iteration: 2900 / 6000 [ 48%]  (Sampling)
## Chain 4 Iteration: 3000 / 6000 [ 50%]  (Sampling)
## Chain 4 Iteration: 3100 / 6000 [ 51%]  (Sampling)
## Chain 3 Iteration: 1100 / 6000 [ 18%]  (Sampling)
## Chain 4 Iteration: 3200 / 6000 [ 53%]  (Sampling)
## Chain 1 Iteration: 1100 / 6000 [ 18%]  (Sampling)
## Chain 2 Iteration: 1100 / 6000 [ 18%]  (Sampling)
## Chain 4 Iteration: 3300 / 6000 [ 55%]  (Sampling)
## Chain 4 Iteration: 3400 / 6000 [ 56%]  (Sampling)
## Chain 4 Iteration: 3500 / 6000 [ 58%]  (Sampling)
## Chain 3 Iteration: 1200 / 6000 [ 20%]  (Sampling)
```

```
## Chain 4 Iteration: 3600 / 6000 [ 60%]  (Sampling)
## Chain 1 Iteration: 1200 / 6000 [ 20%]  (Sampling)
## Chain 2 Iteration: 1200 / 6000 [ 20%]  (Sampling)
## Chain 4 Iteration: 3700 / 6000 [ 61%]  (Sampling)
## Chain 4 Iteration: 3800 / 6000 [ 63%]  (Sampling)
## Chain 4 Iteration: 3900 / 6000 [ 65%]  (Sampling)
## Chain 3 Iteration: 1300 / 6000 [ 21%]  (Sampling)
## Chain 4 Iteration: 4000 / 6000 [ 66%]  (Sampling)
## Chain 1 Iteration: 1300 / 6000 [ 21%]  (Sampling)
## Chain 2 Iteration: 1300 / 6000 [ 21%]  (Sampling)
## Chain 4 Iteration: 4100 / 6000 [ 68%]  (Sampling)
## Chain 4 Iteration: 4200 / 6000 [ 70%]  (Sampling)
## Chain 4 Iteration: 4300 / 6000 [ 71%]  (Sampling)
## Chain 3 Iteration: 1400 / 6000 [ 23%]  (Sampling)
## Chain 4 Iteration: 4400 / 6000 [ 73%]  (Sampling)
## Chain 1 Iteration: 1400 / 6000 [ 23%]  (Sampling)
## Chain 4 Iteration: 4500 / 6000 [ 75%]  (Sampling)
## Chain 2 Iteration: 1400 / 6000 [ 23%]  (Sampling)
## Chain 4 Iteration: 4600 / 6000 [ 76%]  (Sampling)
## Chain 4 Iteration: 4700 / 6000 [ 78%]  (Sampling)
## Chain 4 Iteration: 4800 / 6000 [ 80%]  (Sampling)
## Chain 3 Iteration: 1500 / 6000 [ 25%]  (Sampling)
## Chain 1 Iteration: 1500 / 6000 [ 25%]  (Sampling)
## Chain 4 Iteration: 4900 / 6000 [ 81%]  (Sampling)
## Chain 2 Iteration: 1500 / 6000 [ 25%]  (Sampling)
## Chain 4 Iteration: 5000 / 6000 [ 83%]  (Sampling)
## Chain 4 Iteration: 5100 / 6000 [ 85%]  (Sampling)
## Chain 4 Iteration: 5200 / 6000 [ 86%]  (Sampling)
## Chain 3 Iteration: 1600 / 6000 [ 26%]  (Sampling)
## Chain 4 Iteration: 5300 / 6000 [ 88%]  (Sampling)
## Chain 1 Iteration: 1600 / 6000 [ 26%]  (Sampling)
## Chain 2 Iteration: 1600 / 6000 [ 26%]  (Sampling)
## Chain 4 Iteration: 5400 / 6000 [ 90%]  (Sampling)
## Chain 4 Iteration: 5500 / 6000 [ 91%]  (Sampling)
## Chain 4 Iteration: 5600 / 6000 [ 93%]  (Sampling)
## Chain 3 Iteration: 1700 / 6000 [ 28%]  (Sampling)
## Chain 4 Iteration: 5700 / 6000 [ 95%]  (Sampling)
## Chain 1 Iteration: 1700 / 6000 [ 28%]  (Sampling)
## Chain 2 Iteration: 1700 / 6000 [ 28%]  (Sampling)
## Chain 4 Iteration: 5800 / 6000 [ 96%]  (Sampling)
## Chain 4 Iteration: 5900 / 6000 [ 98%]  (Sampling)
## Chain 4 Iteration: 6000 / 6000 [100%]  (Sampling)
## Chain 4 finished in 116.4 seconds.
## Chain 3 Iteration: 1800 / 6000 [ 30%]  (Sampling)
## Chain 1 Iteration: 1800 / 6000 [ 30%]  (Sampling)
## Chain 2 Iteration: 1800 / 6000 [ 30%]  (Sampling)
## Chain 3 Iteration: 1900 / 6000 [ 31%]  (Sampling)
## Chain 1 Iteration: 1900 / 6000 [ 31%]  (Sampling)
## Chain 2 Iteration: 1900 / 6000 [ 31%]  (Sampling)
## Chain 3 Iteration: 2000 / 6000 [ 33%]  (Sampling)
## Chain 1 Iteration: 2000 / 6000 [ 33%]  (Sampling)
## Chain 2 Iteration: 2000 / 6000 [ 33%]  (Sampling)
## Chain 3 Iteration: 2100 / 6000 [ 35%]  (Sampling)
## Chain 1 Iteration: 2100 / 6000 [ 35%]  (Sampling)
```

```
## Chain 2 Iteration: 2100 / 6000 [ 35%]  (Sampling)
## Chain 3 Iteration: 2200 / 6000 [ 36%]  (Sampling)
## Chain 1 Iteration: 2200 / 6000 [ 36%]  (Sampling)
## Chain 2 Iteration: 2200 / 6000 [ 36%]  (Sampling)
## Chain 3 Iteration: 2300 / 6000 [ 38%]  (Sampling)
## Chain 1 Iteration: 2300 / 6000 [ 38%]  (Sampling)
## Chain 2 Iteration: 2300 / 6000 [ 38%]  (Sampling)
## Chain 3 Iteration: 2400 / 6000 [ 40%]  (Sampling)
## Chain 1 Iteration: 2400 / 6000 [ 40%]  (Sampling)
## Chain 2 Iteration: 2400 / 6000 [ 40%]  (Sampling)
## Chain 3 Iteration: 2500 / 6000 [ 41%]  (Sampling)
## Chain 1 Iteration: 2500 / 6000 [ 41%]  (Sampling)
## Chain 2 Iteration: 2500 / 6000 [ 41%]  (Sampling)
## Chain 3 Iteration: 2600 / 6000 [ 43%]  (Sampling)
## Chain 1 Iteration: 2600 / 6000 [ 43%]  (Sampling)
## Chain 2 Iteration: 2600 / 6000 [ 43%]  (Sampling)
## Chain 3 Iteration: 2700 / 6000 [ 45%]  (Sampling)
## Chain 1 Iteration: 2700 / 6000 [ 45%]  (Sampling)
## Chain 2 Iteration: 2700 / 6000 [ 45%]  (Sampling)
## Chain 3 Iteration: 2800 / 6000 [ 46%]  (Sampling)
## Chain 1 Iteration: 2800 / 6000 [ 46%]  (Sampling)
## Chain 2 Iteration: 2800 / 6000 [ 46%]  (Sampling)
## Chain 3 Iteration: 2900 / 6000 [ 48%]  (Sampling)
## Chain 1 Iteration: 2900 / 6000 [ 48%]  (Sampling)
## Chain 2 Iteration: 2900 / 6000 [ 48%]  (Sampling)
## Chain 3 Iteration: 3000 / 6000 [ 50%]  (Sampling)
## Chain 1 Iteration: 3000 / 6000 [ 50%]  (Sampling)
## Chain 2 Iteration: 3000 / 6000 [ 50%]  (Sampling)
## Chain 3 Iteration: 3100 / 6000 [ 51%]  (Sampling)
## Chain 1 Iteration: 3100 / 6000 [ 51%]  (Sampling)
## Chain 2 Iteration: 3100 / 6000 [ 51%]  (Sampling)
## Chain 3 Iteration: 3200 / 6000 [ 53%]  (Sampling)
## Chain 1 Iteration: 3200 / 6000 [ 53%]  (Sampling)
## Chain 2 Iteration: 3200 / 6000 [ 53%]  (Sampling)
## Chain 3 Iteration: 3300 / 6000 [ 55%]  (Sampling)
## Chain 1 Iteration: 3300 / 6000 [ 55%]  (Sampling)
## Chain 2 Iteration: 3300 / 6000 [ 55%]  (Sampling)
## Chain 3 Iteration: 3400 / 6000 [ 56%]  (Sampling)
## Chain 1 Iteration: 3400 / 6000 [ 56%]  (Sampling)
## Chain 2 Iteration: 3400 / 6000 [ 56%]  (Sampling)
## Chain 3 Iteration: 3500 / 6000 [ 58%]  (Sampling)
## Chain 1 Iteration: 3500 / 6000 [ 58%]  (Sampling)
## Chain 2 Iteration: 3500 / 6000 [ 58%]  (Sampling)
## Chain 3 Iteration: 3600 / 6000 [ 60%]  (Sampling)
## Chain 1 Iteration: 3600 / 6000 [ 60%]  (Sampling)
## Chain 2 Iteration: 3600 / 6000 [ 60%]  (Sampling)
## Chain 3 Iteration: 3700 / 6000 [ 61%]  (Sampling)
## Chain 1 Iteration: 3700 / 6000 [ 61%]  (Sampling)
## Chain 2 Iteration: 3700 / 6000 [ 61%]  (Sampling)
## Chain 3 Iteration: 3800 / 6000 [ 63%]  (Sampling)
## Chain 1 Iteration: 3800 / 6000 [ 63%]  (Sampling)
## Chain 2 Iteration: 3800 / 6000 [ 63%]  (Sampling)
## Chain 3 Iteration: 3900 / 6000 [ 65%]  (Sampling)
## Chain 1 Iteration: 3900 / 6000 [ 65%]  (Sampling)
```

```
## Chain 2 Iteration: 3900 / 6000 [ 65%]  (Sampling)
## Chain 3 Iteration: 4000 / 6000 [ 66%]  (Sampling)
## Chain 1 Iteration: 4000 / 6000 [ 66%]  (Sampling)
## Chain 2 Iteration: 4000 / 6000 [ 66%]  (Sampling)
## Chain 3 Iteration: 4100 / 6000 [ 68%]  (Sampling)
## Chain 1 Iteration: 4100 / 6000 [ 68%]  (Sampling)
## Chain 2 Iteration: 4100 / 6000 [ 68%]  (Sampling)
## Chain 3 Iteration: 4200 / 6000 [ 70%]  (Sampling)
## Chain 1 Iteration: 4200 / 6000 [ 70%]  (Sampling)
## Chain 2 Iteration: 4200 / 6000 [ 70%]  (Sampling)
## Chain 3 Iteration: 4300 / 6000 [ 71%]  (Sampling)
## Chain 1 Iteration: 4300 / 6000 [ 71%]  (Sampling)
## Chain 2 Iteration: 4300 / 6000 [ 71%]  (Sampling)
## Chain 3 Iteration: 4400 / 6000 [ 73%]  (Sampling)
## Chain 1 Iteration: 4400 / 6000 [ 73%]  (Sampling)
## Chain 2 Iteration: 4400 / 6000 [ 73%]  (Sampling)
## Chain 3 Iteration: 4500 / 6000 [ 75%]  (Sampling)
## Chain 1 Iteration: 4500 / 6000 [ 75%]  (Sampling)
## Chain 2 Iteration: 4500 / 6000 [ 75%]  (Sampling)
## Chain 3 Iteration: 4600 / 6000 [ 76%]  (Sampling)
## Chain 1 Iteration: 4600 / 6000 [ 76%]  (Sampling)
## Chain 2 Iteration: 4600 / 6000 [ 76%]  (Sampling)
## Chain 3 Iteration: 4700 / 6000 [ 78%]  (Sampling)
## Chain 1 Iteration: 4700 / 6000 [ 78%]  (Sampling)
## Chain 2 Iteration: 4700 / 6000 [ 78%]  (Sampling)
## Chain 3 Iteration: 4800 / 6000 [ 80%]  (Sampling)
## Chain 1 Iteration: 4800 / 6000 [ 80%]  (Sampling)
## Chain 2 Iteration: 4800 / 6000 [ 80%]  (Sampling)
## Chain 3 Iteration: 4900 / 6000 [ 81%]  (Sampling)
## Chain 1 Iteration: 4900 / 6000 [ 81%]  (Sampling)
## Chain 2 Iteration: 4900 / 6000 [ 81%]  (Sampling)
## Chain 3 Iteration: 5000 / 6000 [ 83%]  (Sampling)
## Chain 1 Iteration: 5000 / 6000 [ 83%]  (Sampling)
## Chain 2 Iteration: 5000 / 6000 [ 83%]  (Sampling)
## Chain 3 Iteration: 5100 / 6000 [ 85%]  (Sampling)
## Chain 1 Iteration: 5100 / 6000 [ 85%]  (Sampling)
## Chain 2 Iteration: 5100 / 6000 [ 85%]  (Sampling)
## Chain 3 Iteration: 5200 / 6000 [ 86%]  (Sampling)
## Chain 1 Iteration: 5200 / 6000 [ 86%]  (Sampling)
## Chain 2 Iteration: 5200 / 6000 [ 86%]  (Sampling)
## Chain 3 Iteration: 5300 / 6000 [ 88%]  (Sampling)
## Chain 1 Iteration: 5300 / 6000 [ 88%]  (Sampling)
## Chain 2 Iteration: 5300 / 6000 [ 88%]  (Sampling)
## Chain 3 Iteration: 5400 / 6000 [ 90%]  (Sampling)
## Chain 1 Iteration: 5400 / 6000 [ 90%]  (Sampling)
## Chain 2 Iteration: 5400 / 6000 [ 90%]  (Sampling)
## Chain 3 Iteration: 5500 / 6000 [ 91%]  (Sampling)
## Chain 1 Iteration: 5500 / 6000 [ 91%]  (Sampling)
## Chain 2 Iteration: 5500 / 6000 [ 91%]  (Sampling)
## Chain 3 Iteration: 5600 / 6000 [ 93%]  (Sampling)
## Chain 1 Iteration: 5600 / 6000 [ 93%]  (Sampling)
## Chain 2 Iteration: 5600 / 6000 [ 93%]  (Sampling)
## Chain 3 Iteration: 5700 / 6000 [ 95%]  (Sampling)
## Chain 1 Iteration: 5700 / 6000 [ 95%]  (Sampling)
```

```
## Chain 2 Iteration: 5700 / 6000 [ 95%]  (Sampling)
## Chain 3 Iteration: 5800 / 6000 [ 96%]  (Sampling)
## Chain 1 Iteration: 5800 / 6000 [ 96%]  (Sampling)
## Chain 2 Iteration: 5800 / 6000 [ 96%]  (Sampling)
## Chain 3 Iteration: 5900 / 6000 [ 98%]  (Sampling)
## Chain 1 Iteration: 5900 / 6000 [ 98%]  (Sampling)
## Chain 2 Iteration: 5900 / 6000 [ 98%]  (Sampling)
## Chain 3 Iteration: 6000 / 6000 [100%]  (Sampling)
## Chain 3 finished in 402.9 seconds.
## Chain 1 Iteration: 6000 / 6000 [100%]  (Sampling)
## Chain 1 finished in 404.9 seconds.
## Chain 2 Iteration: 6000 / 6000 [100%]  (Sampling)
## Chain 2 finished in 405.0 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 332.3 seconds.
## Total execution time: 405.1 seconds.

## Running MCMC with 4 parallel chains...
##
## Chain 1 Iteration:    1 / 6000 [  0%]  (Warmup)
## Chain 2 Iteration:    1 / 6000 [  0%]  (Warmup)
## Chain 3 Iteration:    1 / 6000 [  0%]  (Warmup)
## Chain 4 Iteration:    1 / 6000 [  0%]  (Warmup)
## Chain 2 Iteration:  100 / 6000 [  1%]  (Warmup)
## Chain 4 Iteration:  100 / 6000 [  1%]  (Warmup)
## Chain 1 Iteration:  100 / 6000 [  1%]  (Warmup)
## Chain 3 Iteration:  100 / 6000 [  1%]  (Warmup)
## Chain 2 Iteration:  200 / 6000 [  3%]  (Warmup)
## Chain 4 Iteration:  200 / 6000 [  3%]  (Warmup)
## Chain 1 Iteration:  200 / 6000 [  3%]  (Warmup)
## Chain 3 Iteration:  200 / 6000 [  3%]  (Warmup)
## Chain 2 Iteration:  300 / 6000 [  5%]  (Warmup)
## Chain 4 Iteration:  300 / 6000 [  5%]  (Warmup)
## Chain 1 Iteration:  300 / 6000 [  5%]  (Warmup)
## Chain 3 Iteration:  300 / 6000 [  5%]  (Warmup)
## Chain 2 Iteration:  400 / 6000 [  6%]  (Warmup)
## Chain 4 Iteration:  400 / 6000 [  6%]  (Warmup)
## Chain 1 Iteration:  400 / 6000 [  6%]  (Warmup)
## Chain 3 Iteration:  400 / 6000 [  6%]  (Warmup)
## Chain 4 Iteration:  500 / 6000 [  8%]  (Warmup)
## Chain 2 Iteration:  500 / 6000 [  8%]  (Warmup)
## Chain 1 Iteration:  500 / 6000 [  8%]  (Warmup)
## Chain 3 Iteration:  500 / 6000 [  8%]  (Warmup)
## Chain 4 Iteration:  600 / 6000 [ 10%]  (Warmup)
## Chain 2 Iteration:  600 / 6000 [ 10%]  (Warmup)
## Chain 1 Iteration:  600 / 6000 [ 10%]  (Warmup)
## Chain 3 Iteration:  600 / 6000 [ 10%]  (Warmup)
## Chain 4 Iteration:  700 / 6000 [ 11%]  (Warmup)
## Chain 2 Iteration:  700 / 6000 [ 11%]  (Warmup)
## Chain 1 Iteration:  700 / 6000 [ 11%]  (Warmup)
## Chain 3 Iteration:  700 / 6000 [ 11%]  (Warmup)
## Chain 4 Iteration:  800 / 6000 [ 13%]  (Warmup)
## Chain 2 Iteration:  800 / 6000 [ 13%]  (Warmup)
```

```
## Chain 1 Iteration:  800 / 6000 [ 13%]  (Warmup)
## Chain 3 Iteration:  800 / 6000 [ 13%]  (Warmup)
## Chain 4 Iteration:  900 / 6000 [ 15%]  (Warmup)
## Chain 2 Iteration:  900 / 6000 [ 15%]  (Warmup)
## Chain 1 Iteration:  900 / 6000 [ 15%]  (Warmup)
## Chain 3 Iteration:  900 / 6000 [ 15%]  (Warmup)
## Chain 4 Iteration: 1000 / 6000 [ 16%]  (Warmup)
## Chain 4 Iteration: 1001 / 6000 [ 16%]  (Sampling)
## Chain 4 Iteration: 1100 / 6000 [ 18%]  (Sampling)
## Chain 2 Iteration: 1000 / 6000 [ 16%]  (Warmup)
## Chain 2 Iteration: 1001 / 6000 [ 16%]  (Sampling)
## Chain 1 Iteration: 1000 / 6000 [ 16%]  (Warmup)
## Chain 1 Iteration: 1001 / 6000 [ 16%]  (Sampling)
## Chain 3 Iteration: 1000 / 6000 [ 16%]  (Warmup)
## Chain 3 Iteration: 1001 / 6000 [ 16%]  (Sampling)
## Chain 4 Iteration: 1200 / 6000 [ 20%]  (Sampling)
## Chain 2 Iteration: 1100 / 6000 [ 18%]  (Sampling)
## Chain 4 Iteration: 1300 / 6000 [ 21%]  (Sampling)
## Chain 1 Iteration: 1100 / 6000 [ 18%]  (Sampling)
## Chain 3 Iteration: 1100 / 6000 [ 18%]  (Sampling)
## Chain 4 Iteration: 1400 / 6000 [ 23%]  (Sampling)
## Chain 2 Iteration: 1200 / 6000 [ 20%]  (Sampling)
## Chain 4 Iteration: 1500 / 6000 [ 25%]  (Sampling)
## Chain 1 Iteration: 1200 / 6000 [ 20%]  (Sampling)
## Chain 3 Iteration: 1200 / 6000 [ 20%]  (Sampling)
## Chain 4 Iteration: 1600 / 6000 [ 26%]  (Sampling)
## Chain 2 Iteration: 1300 / 6000 [ 21%]  (Sampling)
## Chain 4 Iteration: 1700 / 6000 [ 28%]  (Sampling)
## Chain 4 Iteration: 1800 / 6000 [ 30%]  (Sampling)
## Chain 2 Iteration: 1400 / 6000 [ 23%]  (Sampling)
## Chain 1 Iteration: 1300 / 6000 [ 21%]  (Sampling)
## Chain 3 Iteration: 1300 / 6000 [ 21%]  (Sampling)
## Chain 4 Iteration: 1900 / 6000 [ 31%]  (Sampling)
## Chain 2 Iteration: 1500 / 6000 [ 25%]  (Sampling)
## Chain 4 Iteration: 2000 / 6000 [ 33%]  (Sampling)
## Chain 1 Iteration: 1400 / 6000 [ 23%]  (Sampling)
## Chain 3 Iteration: 1400 / 6000 [ 23%]  (Sampling)
## Chain 4 Iteration: 2100 / 6000 [ 35%]  (Sampling)
## Chain 2 Iteration: 1600 / 6000 [ 26%]  (Sampling)
## Chain 4 Iteration: 2200 / 6000 [ 36%]  (Sampling)
## Chain 1 Iteration: 1500 / 6000 [ 25%]  (Sampling)
## Chain 3 Iteration: 1500 / 6000 [ 25%]  (Sampling)
## Chain 2 Iteration: 1700 / 6000 [ 28%]  (Sampling)
## Chain 4 Iteration: 2300 / 6000 [ 38%]  (Sampling)
## Chain 4 Iteration: 2400 / 6000 [ 40%]  (Sampling)
## Chain 1 Iteration: 1600 / 6000 [ 26%]  (Sampling)
## Chain 3 Iteration: 1600 / 6000 [ 26%]  (Sampling)
## Chain 2 Iteration: 1800 / 6000 [ 30%]  (Sampling)
## Chain 4 Iteration: 2500 / 6000 [ 41%]  (Sampling)
## Chain 4 Iteration: 2600 / 6000 [ 43%]  (Sampling)
## Chain 2 Iteration: 1900 / 6000 [ 31%]  (Sampling)
## Chain 1 Iteration: 1700 / 6000 [ 28%]  (Sampling)
## Chain 3 Iteration: 1700 / 6000 [ 28%]  (Sampling)
## Chain 4 Iteration: 2700 / 6000 [ 45%]  (Sampling)
```

```
## Chain 2 Iteration: 2000 / 6000 [ 33%]  (Sampling)
## Chain 4 Iteration: 2800 / 6000 [ 46%]  (Sampling)
## Chain 1 Iteration: 1800 / 6000 [ 30%]  (Sampling)
## Chain 3 Iteration: 1800 / 6000 [ 30%]  (Sampling)
## Chain 4 Iteration: 2900 / 6000 [ 48%]  (Sampling)
## Chain 2 Iteration: 2100 / 6000 [ 35%]  (Sampling)
## Chain 4 Iteration: 3000 / 6000 [ 50%]  (Sampling)
## Chain 1 Iteration: 1900 / 6000 [ 31%]  (Sampling)
## Chain 3 Iteration: 1900 / 6000 [ 31%]  (Sampling)
## Chain 4 Iteration: 3100 / 6000 [ 51%]  (Sampling)
## Chain 2 Iteration: 2200 / 6000 [ 36%]  (Sampling)
## Chain 4 Iteration: 3200 / 6000 [ 53%]  (Sampling)
## Chain 2 Iteration: 2300 / 6000 [ 38%]  (Sampling)
## Chain 1 Iteration: 2000 / 6000 [ 33%]  (Sampling)
## Chain 3 Iteration: 2000 / 6000 [ 33%]  (Sampling)
## Chain 4 Iteration: 3300 / 6000 [ 55%]  (Sampling)
## Chain 4 Iteration: 3400 / 6000 [ 56%]  (Sampling)
## Chain 2 Iteration: 2400 / 6000 [ 40%]  (Sampling)
## Chain 4 Iteration: 3500 / 6000 [ 58%]  (Sampling)
## Chain 1 Iteration: 2100 / 6000 [ 35%]  (Sampling)
## Chain 3 Iteration: 2100 / 6000 [ 35%]  (Sampling)
## Chain 2 Iteration: 2500 / 6000 [ 41%]  (Sampling)
## Chain 4 Iteration: 3600 / 6000 [ 60%]  (Sampling)
## Chain 4 Iteration: 3700 / 6000 [ 61%]  (Sampling)
## Chain 1 Iteration: 2200 / 6000 [ 36%]  (Sampling)
## Chain 3 Iteration: 2200 / 6000 [ 36%]  (Sampling)
## Chain 2 Iteration: 2600 / 6000 [ 43%]  (Sampling)
## Chain 4 Iteration: 3800 / 6000 [ 63%]  (Sampling)
## Chain 4 Iteration: 3900 / 6000 [ 65%]  (Sampling)
## Chain 2 Iteration: 2700 / 6000 [ 45%]  (Sampling)
## Chain 1 Iteration: 2300 / 6000 [ 38%]  (Sampling)
## Chain 3 Iteration: 2300 / 6000 [ 38%]  (Sampling)
## Chain 4 Iteration: 4000 / 6000 [ 66%]  (Sampling)
## Chain 2 Iteration: 2800 / 6000 [ 46%]  (Sampling)
## Chain 4 Iteration: 4100 / 6000 [ 68%]  (Sampling)
## Chain 1 Iteration: 2400 / 6000 [ 40%]  (Sampling)
## Chain 3 Iteration: 2400 / 6000 [ 40%]  (Sampling)
## Chain 4 Iteration: 4200 / 6000 [ 70%]  (Sampling)
## Chain 2 Iteration: 2900 / 6000 [ 48%]  (Sampling)
## Chain 4 Iteration: 4300 / 6000 [ 71%]  (Sampling)
## Chain 1 Iteration: 2500 / 6000 [ 41%]  (Sampling)
## Chain 3 Iteration: 2500 / 6000 [ 41%]  (Sampling)
## Chain 4 Iteration: 4400 / 6000 [ 73%]  (Sampling)
## Chain 2 Iteration: 3000 / 6000 [ 50%]  (Sampling)
## Chain 4 Iteration: 4500 / 6000 [ 75%]  (Sampling)
## Chain 1 Iteration: 2600 / 6000 [ 43%]  (Sampling)
## Chain 3 Iteration: 2600 / 6000 [ 43%]  (Sampling)
## Chain 2 Iteration: 3100 / 6000 [ 51%]  (Sampling)
## Chain 4 Iteration: 4600 / 6000 [ 76%]  (Sampling)
## Chain 4 Iteration: 4700 / 6000 [ 78%]  (Sampling)
## Chain 2 Iteration: 3200 / 6000 [ 53%]  (Sampling)
## Chain 4 Iteration: 4800 / 6000 [ 80%]  (Sampling)
## Chain 1 Iteration: 2700 / 6000 [ 45%]  (Sampling)
## Chain 3 Iteration: 2700 / 6000 [ 45%]  (Sampling)
```

```
## Chain 4 Iteration: 4900 / 6000 [ 81%]  (Sampling)
## Chain 2 Iteration: 3300 / 6000 [ 55%]  (Sampling)
## Chain 4 Iteration: 5000 / 6000 [ 83%]  (Sampling)
## Chain 1 Iteration: 2800 / 6000 [ 46%]  (Sampling)
## Chain 3 Iteration: 2800 / 6000 [ 46%]  (Sampling)
## Chain 4 Iteration: 5100 / 6000 [ 85%]  (Sampling)
## Chain 2 Iteration: 3400 / 6000 [ 56%]  (Sampling)
## Chain 4 Iteration: 5200 / 6000 [ 86%]  (Sampling)
## Chain 1 Iteration: 2900 / 6000 [ 48%]  (Sampling)
## Chain 3 Iteration: 2900 / 6000 [ 48%]  (Sampling)
## Chain 2 Iteration: 3500 / 6000 [ 58%]  (Sampling)
## Chain 4 Iteration: 5300 / 6000 [ 88%]  (Sampling)
## Chain 4 Iteration: 5400 / 6000 [ 90%]  (Sampling)
## Chain 2 Iteration: 3600 / 6000 [ 60%]  (Sampling)
## Chain 1 Iteration: 3000 / 6000 [ 50%]  (Sampling)
## Chain 3 Iteration: 3000 / 6000 [ 50%]  (Sampling)
## Chain 4 Iteration: 5500 / 6000 [ 91%]  (Sampling)
## Chain 4 Iteration: 5600 / 6000 [ 93%]  (Sampling)
## Chain 2 Iteration: 3700 / 6000 [ 61%]  (Sampling)
## Chain 1 Iteration: 3100 / 6000 [ 51%]  (Sampling)
## Chain 3 Iteration: 3100 / 6000 [ 51%]  (Sampling)
## Chain 4 Iteration: 5700 / 6000 [ 95%]  (Sampling)
## Chain 2 Iteration: 3800 / 6000 [ 63%]  (Sampling)
## Chain 4 Iteration: 5800 / 6000 [ 96%]  (Sampling)
## Chain 1 Iteration: 3200 / 6000 [ 53%]  (Sampling)
## Chain 3 Iteration: 3200 / 6000 [ 53%]  (Sampling)
## Chain 4 Iteration: 5900 / 6000 [ 98%]  (Sampling)
## Chain 2 Iteration: 3900 / 6000 [ 65%]  (Sampling)
## Chain 4 Iteration: 6000 / 6000 [100%]  (Sampling)
## Chain 4 finished in 217.7 seconds.
## Chain 1 Iteration: 3300 / 6000 [ 55%]  (Sampling)
## Chain 3 Iteration: 3300 / 6000 [ 55%]  (Sampling)
## Chain 2 Iteration: 4000 / 6000 [ 66%]  (Sampling)
## Chain 2 Iteration: 4100 / 6000 [ 68%]  (Sampling)
## Chain 1 Iteration: 3400 / 6000 [ 56%]  (Sampling)
## Chain 3 Iteration: 3400 / 6000 [ 56%]  (Sampling)
## Chain 2 Iteration: 4200 / 6000 [ 70%]  (Sampling)
## Chain 1 Iteration: 3500 / 6000 [ 58%]  (Sampling)
## Chain 3 Iteration: 3500 / 6000 [ 58%]  (Sampling)
## Chain 2 Iteration: 4300 / 6000 [ 71%]  (Sampling)
## Chain 1 Iteration: 3600 / 6000 [ 60%]  (Sampling)
## Chain 3 Iteration: 3600 / 6000 [ 60%]  (Sampling)
## Chain 2 Iteration: 4400 / 6000 [ 73%]  (Sampling)
## Chain 2 Iteration: 4500 / 6000 [ 75%]  (Sampling)
## Chain 1 Iteration: 3700 / 6000 [ 61%]  (Sampling)
## Chain 3 Iteration: 3700 / 6000 [ 61%]  (Sampling)
## Chain 2 Iteration: 4600 / 6000 [ 76%]  (Sampling)
## Chain 1 Iteration: 3800 / 6000 [ 63%]  (Sampling)
## Chain 3 Iteration: 3800 / 6000 [ 63%]  (Sampling)
## Chain 2 Iteration: 4700 / 6000 [ 78%]  (Sampling)
## Chain 1 Iteration: 3900 / 6000 [ 65%]  (Sampling)
## Chain 3 Iteration: 3900 / 6000 [ 65%]  (Sampling)
## Chain 2 Iteration: 4800 / 6000 [ 80%]  (Sampling)
## Chain 1 Iteration: 4000 / 6000 [ 66%]  (Sampling)
```

```
## Chain 3 Iteration: 4000 / 6000 [ 66%]  (Sampling)
## Chain 2 Iteration: 4900 / 6000 [ 81%]  (Sampling)
## Chain 2 Iteration: 5000 / 6000 [ 83%]  (Sampling)
## Chain 1 Iteration: 4100 / 6000 [ 68%]  (Sampling)
## Chain 3 Iteration: 4100 / 6000 [ 68%]  (Sampling)
## Chain 2 Iteration: 5100 / 6000 [ 85%]  (Sampling)
## Chain 1 Iteration: 4200 / 6000 [ 70%]  (Sampling)
## Chain 3 Iteration: 4200 / 6000 [ 70%]  (Sampling)
## Chain 2 Iteration: 5200 / 6000 [ 86%]  (Sampling)
## Chain 1 Iteration: 4300 / 6000 [ 71%]  (Sampling)
## Chain 3 Iteration: 4300 / 6000 [ 71%]  (Sampling)
## Chain 2 Iteration: 5300 / 6000 [ 88%]  (Sampling)
## Chain 2 Iteration: 5400 / 6000 [ 90%]  (Sampling)
## Chain 1 Iteration: 4400 / 6000 [ 73%]  (Sampling)
## Chain 3 Iteration: 4400 / 6000 [ 73%]  (Sampling)
## Chain 2 Iteration: 5500 / 6000 [ 91%]  (Sampling)
## Chain 1 Iteration: 4500 / 6000 [ 75%]  (Sampling)
## Chain 3 Iteration: 4500 / 6000 [ 75%]  (Sampling)
## Chain 2 Iteration: 5600 / 6000 [ 93%]  (Sampling)
## Chain 1 Iteration: 4600 / 6000 [ 76%]  (Sampling)
## Chain 3 Iteration: 4600 / 6000 [ 76%]  (Sampling)
## Chain 2 Iteration: 5700 / 6000 [ 95%]  (Sampling)
## Chain 2 Iteration: 5800 / 6000 [ 96%]  (Sampling)
## Chain 1 Iteration: 4700 / 6000 [ 78%]  (Sampling)
## Chain 3 Iteration: 4700 / 6000 [ 78%]  (Sampling)
## Chain 2 Iteration: 5900 / 6000 [ 98%]  (Sampling)
## Chain 1 Iteration: 4800 / 6000 [ 80%]  (Sampling)
## Chain 3 Iteration: 4800 / 6000 [ 80%]  (Sampling)
## Chain 2 Iteration: 6000 / 6000 [100%]  (Sampling)
## Chain 2 finished in 326.3 seconds.
## Chain 1 Iteration: 4900 / 6000 [ 81%]  (Sampling)
## Chain 3 Iteration: 4900 / 6000 [ 81%]  (Sampling)
## Chain 1 Iteration: 5000 / 6000 [ 83%]  (Sampling)
## Chain 3 Iteration: 5000 / 6000 [ 83%]  (Sampling)
## Chain 1 Iteration: 5100 / 6000 [ 85%]  (Sampling)
## Chain 3 Iteration: 5100 / 6000 [ 85%]  (Sampling)
## Chain 1 Iteration: 5200 / 6000 [ 86%]  (Sampling)
## Chain 3 Iteration: 5200 / 6000 [ 86%]  (Sampling)
## Chain 1 Iteration: 5300 / 6000 [ 88%]  (Sampling)
## Chain 3 Iteration: 5300 / 6000 [ 88%]  (Sampling)
## Chain 1 Iteration: 5400 / 6000 [ 90%]  (Sampling)
## Chain 3 Iteration: 5400 / 6000 [ 90%]  (Sampling)
## Chain 1 Iteration: 5500 / 6000 [ 91%]  (Sampling)
## Chain 3 Iteration: 5500 / 6000 [ 91%]  (Sampling)
## Chain 1 Iteration: 5600 / 6000 [ 93%]  (Sampling)
## Chain 3 Iteration: 5600 / 6000 [ 93%]  (Sampling)
## Chain 1 Iteration: 5700 / 6000 [ 95%]  (Sampling)
## Chain 3 Iteration: 5700 / 6000 [ 95%]  (Sampling)
## Chain 1 Iteration: 5800 / 6000 [ 96%]  (Sampling)
## Chain 3 Iteration: 5800 / 6000 [ 96%]  (Sampling)
## Chain 1 Iteration: 5900 / 6000 [ 98%]  (Sampling)
## Chain 3 Iteration: 5900 / 6000 [ 98%]  (Sampling)
## Chain 1 Iteration: 6000 / 6000 [100%]  (Sampling)
## Chain 3 Iteration: 6000 / 6000 [100%]  (Sampling)
```

```
## Chain 1 finished in 403.1 seconds.
## Chain 3 finished in 403.1 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 337.6 seconds.
## Total execution time: 403.2 seconds.

## Running MCMC with 4 parallel chains...
##
## Chain 1 Iteration:    1 / 6000 [  0%]  (Warmup)
## Chain 2 Iteration:    1 / 6000 [  0%]  (Warmup)
## Chain 3 Iteration:    1 / 6000 [  0%]  (Warmup)
## Chain 4 Iteration:    1 / 6000 [  0%]  (Warmup)
## Chain 2 Iteration:  100 / 6000 [  1%]  (Warmup)
## Chain 3 Iteration:  100 / 6000 [  1%]  (Warmup)
## Chain 1 Iteration:  100 / 6000 [  1%]  (Warmup)
## Chain 4 Iteration:  100 / 6000 [  1%]  (Warmup)
## Chain 3 Iteration:  200 / 6000 [  3%]  (Warmup)
## Chain 2 Iteration:  200 / 6000 [  3%]  (Warmup)
## Chain 1 Iteration:  200 / 6000 [  3%]  (Warmup)
## Chain 4 Iteration:  200 / 6000 [  3%]  (Warmup)
## Chain 3 Iteration:  300 / 6000 [  5%]  (Warmup)
## Chain 1 Iteration:  300 / 6000 [  5%]  (Warmup)
## Chain 2 Iteration:  300 / 6000 [  5%]  (Warmup)
## Chain 4 Iteration:  300 / 6000 [  5%]  (Warmup)
## Chain 3 Iteration:  400 / 6000 [  6%]  (Warmup)
## Chain 1 Iteration:  400 / 6000 [  6%]  (Warmup)
## Chain 4 Iteration:  400 / 6000 [  6%]  (Warmup)
## Chain 2 Iteration:  400 / 6000 [  6%]  (Warmup)
## Chain 3 Iteration:  500 / 6000 [  8%]  (Warmup)
## Chain 3 Iteration:  600 / 6000 [ 10%]  (Warmup)
## Chain 3 Iteration:  700 / 6000 [ 11%]  (Warmup)
## Chain 3 Iteration:  800 / 6000 [ 13%]  (Warmup)
## Chain 3 Iteration:  900 / 6000 [ 15%]  (Warmup)
## Chain 3 Iteration: 1000 / 6000 [ 16%]  (Warmup)
## Chain 2 Iteration:  500 / 6000 [  8%]  (Warmup)
## Chain 3 Iteration: 1001 / 6000 [ 16%]  (Sampling)
## Chain 3 Iteration: 1100 / 6000 [ 18%]  (Sampling)
## Chain 2 Iteration:  600 / 6000 [ 10%]  (Warmup)
## Chain 3 Iteration: 1200 / 6000 [ 20%]  (Sampling)
## Chain 2 Iteration:  700 / 6000 [ 11%]  (Warmup)
## Chain 3 Iteration: 1300 / 6000 [ 21%]  (Sampling)
## Chain 2 Iteration:  800 / 6000 [ 13%]  (Warmup)
## Chain 3 Iteration: 1400 / 6000 [ 23%]  (Sampling)
## Chain 2 Iteration:  900 / 6000 [ 15%]  (Warmup)
## Chain 3 Iteration: 1500 / 6000 [ 25%]  (Sampling)
## Chain 2 Iteration: 1000 / 6000 [ 16%]  (Warmup)
## Chain 2 Iteration: 1001 / 6000 [ 16%]  (Sampling)
## Chain 3 Iteration: 1600 / 6000 [ 26%]  (Sampling)
## Chain 3 Iteration: 1700 / 6000 [ 28%]  (Sampling)
## Chain 2 Iteration: 1100 / 6000 [ 18%]  (Sampling)
## Chain 2 Iteration: 1200 / 6000 [ 20%]  (Sampling)
## Chain 3 Iteration: 1800 / 6000 [ 30%]  (Sampling)
## Chain 2 Iteration: 1300 / 6000 [ 21%]  (Sampling)
```

```
## Chain 3 Iteration: 1900 / 6000 [ 31%]  (Sampling)
## Chain 3 Iteration: 2000 / 6000 [ 33%]  (Sampling)
## Chain 4 Iteration:  500 / 6000 [  8%]  (Warmup)
## Chain 2 Iteration: 1400 / 6000 [ 23%]  (Sampling)
## Chain 3 Iteration: 2100 / 6000 [ 35%]  (Sampling)
## Chain 1 Iteration:  500 / 6000 [  8%]  (Warmup)
## Chain 2 Iteration: 1500 / 6000 [ 25%]  (Sampling)
## Chain 2 Iteration: 1600 / 6000 [ 26%]  (Sampling)
## Chain 3 Iteration: 2200 / 6000 [ 36%]  (Sampling)
## Chain 2 Iteration: 1700 / 6000 [ 28%]  (Sampling)
## Chain 3 Iteration: 2300 / 6000 [ 38%]  (Sampling)
## Chain 3 Iteration: 2400 / 6000 [ 40%]  (Sampling)
## Chain 2 Iteration: 1800 / 6000 [ 30%]  (Sampling)
## Chain 3 Iteration: 2500 / 6000 [ 41%]  (Sampling)
## Chain 2 Iteration: 1900 / 6000 [ 31%]  (Sampling)
## Chain 2 Iteration: 2000 / 6000 [ 33%]  (Sampling)
## Chain 3 Iteration: 2600 / 6000 [ 43%]  (Sampling)
## Chain 3 Iteration: 2700 / 6000 [ 45%]  (Sampling)
## Chain 2 Iteration: 2100 / 6000 [ 35%]  (Sampling)
## Chain 3 Iteration: 2800 / 6000 [ 46%]  (Sampling)
## Chain 2 Iteration: 2200 / 6000 [ 36%]  (Sampling)
## Chain 2 Iteration: 2300 / 6000 [ 38%]  (Sampling)
## Chain 3 Iteration: 2900 / 6000 [ 48%]  (Sampling)
## Chain 3 Iteration: 3000 / 6000 [ 50%]  (Sampling)
## Chain 2 Iteration: 2400 / 6000 [ 40%]  (Sampling)
## Chain 3 Iteration: 3100 / 6000 [ 51%]  (Sampling)
## Chain 2 Iteration: 2500 / 6000 [ 41%]  (Sampling)
## Chain 3 Iteration: 3200 / 6000 [ 53%]  (Sampling)
## Chain 2 Iteration: 2600 / 6000 [ 43%]  (Sampling)
## Chain 2 Iteration: 2700 / 6000 [ 45%]  (Sampling)
## Chain 3 Iteration: 3300 / 6000 [ 55%]  (Sampling)
## Chain 3 Iteration: 3400 / 6000 [ 56%]  (Sampling)
## Chain 2 Iteration: 2800 / 6000 [ 46%]  (Sampling)
## Chain 3 Iteration: 3500 / 6000 [ 58%]  (Sampling)
## Chain 2 Iteration: 2900 / 6000 [ 48%]  (Sampling)
## Chain 2 Iteration: 3000 / 6000 [ 50%]  (Sampling)
## Chain 3 Iteration: 3600 / 6000 [ 60%]  (Sampling)
## Chain 4 Iteration:  600 / 6000 [ 10%]  (Warmup)
## Chain 3 Iteration: 3700 / 6000 [ 61%]  (Sampling)
## Chain 2 Iteration: 3100 / 6000 [ 51%]  (Sampling)
## Chain 3 Iteration: 3800 / 6000 [ 63%]  (Sampling)
## Chain 2 Iteration: 3200 / 6000 [ 53%]  (Sampling)
## Chain 3 Iteration: 3900 / 6000 [ 65%]  (Sampling)
## Chain 2 Iteration: 3300 / 6000 [ 55%]  (Sampling)
## Chain 3 Iteration: 4000 / 6000 [ 66%]  (Sampling)
## Chain 2 Iteration: 3400 / 6000 [ 56%]  (Sampling)
## Chain 3 Iteration: 4100 / 6000 [ 68%]  (Sampling)
## Chain 2 Iteration: 3500 / 6000 [ 58%]  (Sampling)
## Chain 3 Iteration: 4200 / 6000 [ 70%]  (Sampling)
## Chain 2 Iteration: 3600 / 6000 [ 60%]  (Sampling)
## Chain 3 Iteration: 4300 / 6000 [ 71%]  (Sampling)
## Chain 2 Iteration: 3700 / 6000 [ 61%]  (Sampling)
## Chain 3 Iteration: 4400 / 6000 [ 73%]  (Sampling)
## Chain 2 Iteration: 3800 / 6000 [ 63%]  (Sampling)
```

```
## Chain 3 Iteration: 4500 / 6000 [ 75%]  (Sampling)
## Chain 2 Iteration: 3900 / 6000 [ 65%]  (Sampling)
## Chain 3 Iteration: 4600 / 6000 [ 76%]  (Sampling)
## Chain 2 Iteration: 4000 / 6000 [ 66%]  (Sampling)
## Chain 3 Iteration: 4700 / 6000 [ 78%]  (Sampling)
## Chain 2 Iteration: 4100 / 6000 [ 68%]  (Sampling)
## Chain 3 Iteration: 4800 / 6000 [ 80%]  (Sampling)
## Chain 2 Iteration: 4200 / 6000 [ 70%]  (Sampling)
## Chain 3 Iteration: 4900 / 6000 [ 81%]  (Sampling)
## Chain 2 Iteration: 4300 / 6000 [ 71%]  (Sampling)
## Chain 3 Iteration: 5000 / 6000 [ 83%]  (Sampling)
## Chain 2 Iteration: 4400 / 6000 [ 73%]  (Sampling)
## Chain 3 Iteration: 5100 / 6000 [ 85%]  (Sampling)
## Chain 2 Iteration: 4500 / 6000 [ 75%]  (Sampling)
## Chain 1 Iteration:  600 / 6000 [ 10%]  (Warmup)
## Chain 3 Iteration: 5200 / 6000 [ 86%]  (Sampling)
## Chain 2 Iteration: 4600 / 6000 [ 76%]  (Sampling)
## Chain 3 Iteration: 5300 / 6000 [ 88%]  (Sampling)
## Chain 4 Iteration:  700 / 6000 [ 11%]  (Warmup)
## Chain 2 Iteration: 4700 / 6000 [ 78%]  (Sampling)
## Chain 3 Iteration: 5400 / 6000 [ 90%]  (Sampling)
## Chain 2 Iteration: 4800 / 6000 [ 80%]  (Sampling)
## Chain 3 Iteration: 5500 / 6000 [ 91%]  (Sampling)
## Chain 2 Iteration: 4900 / 6000 [ 81%]  (Sampling)
## Chain 3 Iteration: 5600 / 6000 [ 93%]  (Sampling)
## Chain 2 Iteration: 5000 / 6000 [ 83%]  (Sampling)
## Chain 3 Iteration: 5700 / 6000 [ 95%]  (Sampling)
## Chain 2 Iteration: 5100 / 6000 [ 85%]  (Sampling)
## Chain 3 Iteration: 5800 / 6000 [ 96%]  (Sampling)
## Chain 2 Iteration: 5200 / 6000 [ 86%]  (Sampling)
## Chain 3 Iteration: 5900 / 6000 [ 98%]  (Sampling)
## Chain 2 Iteration: 5300 / 6000 [ 88%]  (Sampling)
## Chain 3 Iteration: 6000 / 6000 [100%]  (Sampling)
## Chain 2 Iteration: 5400 / 6000 [ 90%]  (Sampling)
## Chain 3 finished in 24.8 seconds.
## Chain 2 Iteration: 5500 / 6000 [ 91%]  (Sampling)
## Chain 2 Iteration: 5600 / 6000 [ 93%]  (Sampling)
## Chain 2 Iteration: 5700 / 6000 [ 95%]  (Sampling)
## Chain 2 Iteration: 5800 / 6000 [ 96%]  (Sampling)
## Chain 2 Iteration: 5900 / 6000 [ 98%]  (Sampling)
## Chain 2 Iteration: 6000 / 6000 [100%]  (Sampling)
## Chain 2 finished in 26.0 seconds.
## Chain 4 Iteration:  800 / 6000 [ 13%]  (Warmup)
## Chain 1 Iteration:  700 / 6000 [ 11%]  (Warmup)
## Chain 4 Iteration:  900 / 6000 [ 15%]  (Warmup)
## Chain 4 Iteration: 1000 / 6000 [ 16%]  (Warmup)
## Chain 4 Iteration: 1001 / 6000 [ 16%]  (Sampling)
## Chain 4 Iteration: 1100 / 6000 [ 18%]  (Sampling)
## Chain 4 Iteration: 1200 / 6000 [ 20%]  (Sampling)
## Chain 4 Iteration: 1300 / 6000 [ 21%]  (Sampling)
## Chain 4 Iteration: 1400 / 6000 [ 23%]  (Sampling)
## Chain 4 Iteration: 1500 / 6000 [ 25%]  (Sampling)
## Chain 4 Iteration: 1600 / 6000 [ 26%]  (Sampling)
## Chain 4 Iteration: 1700 / 6000 [ 28%]  (Sampling)
```

```
## Chain 4 Iteration: 1800 / 6000 [ 30%]  (Sampling)
## Chain 4 Iteration: 1900 / 6000 [ 31%]  (Sampling)
## Chain 4 Iteration: 2000 / 6000 [ 33%]  (Sampling)
## Chain 4 Iteration: 2100 / 6000 [ 35%]  (Sampling)
## Chain 4 Iteration: 2200 / 6000 [ 36%]  (Sampling)
## Chain 4 Iteration: 2300 / 6000 [ 38%]  (Sampling)
## Chain 1 Iteration:  800 / 6000 [ 13%]  (Warmup)
## Chain 4 Iteration: 2400 / 6000 [ 40%]  (Sampling)
## Chain 4 Iteration: 2500 / 6000 [ 41%]  (Sampling)
## Chain 4 Iteration: 2600 / 6000 [ 43%]  (Sampling)
## Chain 4 Iteration: 2700 / 6000 [ 45%]  (Sampling)
## Chain 4 Iteration: 2800 / 6000 [ 46%]  (Sampling)
## Chain 4 Iteration: 2900 / 6000 [ 48%]  (Sampling)
## Chain 4 Iteration: 3000 / 6000 [ 50%]  (Sampling)
## Chain 4 Iteration: 3100 / 6000 [ 51%]  (Sampling)
## Chain 4 Iteration: 3200 / 6000 [ 53%]  (Sampling)
## Chain 4 Iteration: 3300 / 6000 [ 55%]  (Sampling)
## Chain 4 Iteration: 3400 / 6000 [ 56%]  (Sampling)
## Chain 4 Iteration: 3500 / 6000 [ 58%]  (Sampling)
## Chain 4 Iteration: 3600 / 6000 [ 60%]  (Sampling)
## Chain 4 Iteration: 3700 / 6000 [ 61%]  (Sampling)
## Chain 4 Iteration: 3800 / 6000 [ 63%]  (Sampling)
## Chain 4 Iteration: 3900 / 6000 [ 65%]  (Sampling)
## Chain 4 Iteration: 4000 / 6000 [ 66%]  (Sampling)
## Chain 4 Iteration: 4100 / 6000 [ 68%]  (Sampling)
## Chain 4 Iteration: 4200 / 6000 [ 70%]  (Sampling)
## Chain 4 Iteration: 4300 / 6000 [ 71%]  (Sampling)
## Chain 4 Iteration: 4400 / 6000 [ 73%]  (Sampling)
## Chain 1 Iteration:  900 / 6000 [ 15%]  (Warmup)
## Chain 4 Iteration: 4500 / 6000 [ 75%]  (Sampling)
## Chain 4 Iteration: 4600 / 6000 [ 76%]  (Sampling)
## Chain 4 Iteration: 4700 / 6000 [ 78%]  (Sampling)
## Chain 4 Iteration: 4800 / 6000 [ 80%]  (Sampling)
## Chain 4 Iteration: 4900 / 6000 [ 81%]  (Sampling)
## Chain 4 Iteration: 5000 / 6000 [ 83%]  (Sampling)
## Chain 4 Iteration: 5100 / 6000 [ 85%]  (Sampling)
## Chain 4 Iteration: 5200 / 6000 [ 86%]  (Sampling)
## Chain 4 Iteration: 5300 / 6000 [ 88%]  (Sampling)
## Chain 4 Iteration: 5400 / 6000 [ 90%]  (Sampling)
## Chain 4 Iteration: 5500 / 6000 [ 91%]  (Sampling)
## Chain 1 Iteration: 1000 / 6000 [ 16%]  (Warmup)
## Chain 1 Iteration: 1001 / 6000 [ 16%]  (Sampling)
## Chain 4 Iteration: 5600 / 6000 [ 93%]  (Sampling)
## Chain 1 Iteration: 1100 / 6000 [ 18%]  (Sampling)
## Chain 4 Iteration: 5700 / 6000 [ 95%]  (Sampling)
## Chain 1 Iteration: 1200 / 6000 [ 20%]  (Sampling)
## Chain 1 Iteration: 1300 / 6000 [ 21%]  (Sampling)
## Chain 4 Iteration: 5800 / 6000 [ 96%]  (Sampling)
## Chain 1 Iteration: 1400 / 6000 [ 23%]  (Sampling)
## Chain 4 Iteration: 5900 / 6000 [ 98%]  (Sampling)
## Chain 1 Iteration: 1500 / 6000 [ 25%]  (Sampling)
## Chain 4 Iteration: 6000 / 6000 [100%]  (Sampling)
## Chain 4 finished in 43.7 seconds.
## Chain 1 Iteration: 1600 / 6000 [ 26%]  (Sampling)
```

```
## Chain 1 Iteration: 1700 / 6000 [ 28%]  (Sampling)
## Chain 1 Iteration: 1800 / 6000 [ 30%]  (Sampling)
## Chain 1 Iteration: 1900 / 6000 [ 31%]  (Sampling)
## Chain 1 Iteration: 2000 / 6000 [ 33%]  (Sampling)
## Chain 1 Iteration: 2100 / 6000 [ 35%]  (Sampling)
## Chain 1 Iteration: 2200 / 6000 [ 36%]  (Sampling)
## Chain 1 Iteration: 2300 / 6000 [ 38%]  (Sampling)
## Chain 1 Iteration: 2400 / 6000 [ 40%]  (Sampling)
## Chain 1 Iteration: 2500 / 6000 [ 41%]  (Sampling)
## Chain 1 Iteration: 2600 / 6000 [ 43%]  (Sampling)
## Chain 1 Iteration: 2700 / 6000 [ 45%]  (Sampling)
## Chain 1 Iteration: 2800 / 6000 [ 46%]  (Sampling)
## Chain 1 Iteration: 2900 / 6000 [ 48%]  (Sampling)
## Chain 1 Iteration: 3000 / 6000 [ 50%]  (Sampling)
## Chain 1 Iteration: 3100 / 6000 [ 51%]  (Sampling)
## Chain 1 Iteration: 3200 / 6000 [ 53%]  (Sampling)
## Chain 1 Iteration: 3300 / 6000 [ 55%]  (Sampling)
## Chain 1 Iteration: 3400 / 6000 [ 56%]  (Sampling)
## Chain 1 Iteration: 3500 / 6000 [ 58%]  (Sampling)
## Chain 1 Iteration: 3600 / 6000 [ 60%]  (Sampling)
## Chain 1 Iteration: 3700 / 6000 [ 61%]  (Sampling)
## Chain 1 Iteration: 3800 / 6000 [ 63%]  (Sampling)
## Chain 1 Iteration: 3900 / 6000 [ 65%]  (Sampling)
## Chain 1 Iteration: 4000 / 6000 [ 66%]  (Sampling)
## Chain 1 Iteration: 4100 / 6000 [ 68%]  (Sampling)
## Chain 1 Iteration: 4200 / 6000 [ 70%]  (Sampling)
## Chain 1 Iteration: 4300 / 6000 [ 71%]  (Sampling)
## Chain 1 Iteration: 4400 / 6000 [ 73%]  (Sampling)
## Chain 1 Iteration: 4500 / 6000 [ 75%]  (Sampling)
## Chain 1 Iteration: 4600 / 6000 [ 76%]  (Sampling)
## Chain 1 Iteration: 4700 / 6000 [ 78%]  (Sampling)
## Chain 1 Iteration: 4800 / 6000 [ 80%]  (Sampling)
## Chain 1 Iteration: 4900 / 6000 [ 81%]  (Sampling)
## Chain 1 Iteration: 5000 / 6000 [ 83%]  (Sampling)
## Chain 1 Iteration: 5100 / 6000 [ 85%]  (Sampling)
## Chain 1 Iteration: 5200 / 6000 [ 86%]  (Sampling)
## Chain 1 Iteration: 5300 / 6000 [ 88%]  (Sampling)
## Chain 1 Iteration: 5400 / 6000 [ 90%]  (Sampling)
## Chain 1 Iteration: 5500 / 6000 [ 91%]  (Sampling)
## Chain 1 Iteration: 5600 / 6000 [ 93%]  (Sampling)
## Chain 1 Iteration: 5700 / 6000 [ 95%]  (Sampling)
## Chain 1 Iteration: 5800 / 6000 [ 96%]  (Sampling)
## Chain 1 Iteration: 5900 / 6000 [ 98%]  (Sampling)
## Chain 1 Iteration: 6000 / 6000 [100%]  (Sampling)
## Chain 1 finished in 52.2 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 36.7 seconds.
## Total execution time: 52.3 seconds.
```

Compare models with leave-one-out cross validation

```
##                     elpd_diff se_diff
## int_mod_bay2            0.0       0.0
```

```
## int_mod_bay         -2278.8     109.3
## full_mod_bay        -2619.2     108.6
## intercept_model_bay -5283.3     116.1
```

Intercept model performs best, let's look at the summary:

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: yday ~ year * flow_90 + temp_90 * stream + SLOPE + mean_elevation + (1 | COMID)
##    Data: model_data_final (Number of observations: 3016)
##   Draws: 4 chains, each with iter = 6000; warmup = 1000; thin = 1;
##          total post-warmup draws = 20000
##
## Multilevel Hyperparameters:
## ~COMID (Number of levels: 104)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     3.45      0.53     2.88     4.10 1.08       33      103
##
## Regression Coefficients:
##                       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept               236.74      0.94   235.45   237.81 1.08       33
## year2003                  4.16      0.15     3.87     4.45 1.00     1848
## year2004                 -5.35      0.12    -5.59    -5.11 1.00     3183
## year2005                 -3.84      0.28    -4.39    -3.28 1.00     5084
## flow_90                  -9.61      0.10    -9.82    -9.41 1.00     2291
## temp_90                   5.34      0.18     4.99     5.70 1.00     1600
## streamBeaver              1.14      0.81    -0.45     2.74 1.02      216
## streamBig                 0.47      0.83    -1.14     2.09 1.02      161
## streamCamas               0.44      0.71    -0.99     1.83 1.01      323
## streamElk                 0.09      0.83    -1.58     1.69 1.02      335
## streamLoon                0.78      0.73    -0.68     2.17 1.01      350
## streamMarsh              -1.42      0.75    -2.86     0.10 1.03      181
## streamSulphur             2.92      0.85     1.20     4.50 1.05       68
## SLOPE                    -0.13      0.14    -0.41     0.15 1.02      240
## mean_elevation            2.93      0.38     2.19     3.65 1.02      273
## year2003:flow_90          5.11      0.06     4.99     5.23 1.00     4923
## year2004:flow_90         -4.58      0.12    -4.81    -4.34 1.00     4635
## year2005:flow_90         -1.33      0.25    -1.82    -0.84 1.00     6329
## temp_90:streamBeaver      0.97      0.22     0.54     1.40 1.00     5447
## temp_90:streamBig         1.98      0.14     1.71     2.27 1.00     2997
## temp_90:streamCamas       0.79      0.18     0.45     1.15 1.00     2338
## temp_90:streamElk         1.38      0.17     1.06     1.71 1.00     3296
## temp_90:streamLoon       -0.48      0.25    -0.96    -0.00 1.00     3931
## temp_90:streamMarsh       0.27      0.20    -0.12     0.66 1.00     4824
## temp_90:streamSulphur     0.89      0.22     0.47     1.33 1.00     2158
##                       Tail_ESS
## Intercept                   27
## year2003                  4584
## year2004                  5769
## year2005                  7254
## flow_90                   4074
## temp_90                   3643
## streamBeaver               565
## streamBig                  500
```

```
## streamCamas               1271
## streamElk                  782
## streamLoon                 879
## streamMarsh                288
## streamSulphur              235
## SLOPE                      448
## mean_elevation            1430
## year2003:flow_90          8537
## year2004:flow_90          7545
## year2005:flow_90          6967
## temp_90:streamBeaver     10468
## temp_90:streamBig         6416
## temp_90:streamCamas       4235
## temp_90:streamElk         6098
## temp_90:streamLoon        8072
## temp_90:streamMarsh       8317
## temp_90:streamSulphur     4725
##
## Further Distributional Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.85      0.01     0.83     0.87 1.00    11266    12191
##
## Draws were sampled using sample(hmc). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```